SAP NetWeaver Guides

SAP NetWeaver on AWS

SAP NetWeaver on AWS: SAP NetWeaver Guides

Table of Contents

SAP NetWeaver Guides

This section covers the following guides.

- SAP NetWeaver Environment Setup for Linux on AWS
- SAP NetWeaver on AWS Deployment and Operations Guide for Windows
- Microsoft SQL Server for SAP NetWeaver on AWS Deployment and Operations Guide
- SAP NetWeaver on AWS: high availability configuration for SUSE Linux Enterprise Server (SLES) for SAP applications
- SAP NetWeaver on AWS: high availability configuration for Red Hat Enterprise Linux (RHEL) for SAP applications
- Migrate SAP NetWeaver applications with AWS Migration Hub Orchestrator
- Oracle for SAP NetWeaver on AWS Deployment and Operations Guide for Linux
- SAP ASE for SAP NetWeaver on AWS Deployment and Operations Guide for Linux
- SAP NetWeaver on AWS Automation

Additional SAP on AWS documentation

- General SAP guides
- SAP HANA on AWS
- Databases for SAP applications on AWS
- AWS Launch Wizard for SAP
- AWS Systems Manager for SAP
- AWS SDK for SAP ABAP
- SAP BusinessObjects on AWS
- AWS Migration Hub Orchestrator

SAP NetWeaver Environment Setup for Linux on AWS

SAP specialists, Amazon Web Services

Last updated: August 2022

This guide describes the prerequisites and procedures for setting up an environment on Amazon Web Services prior to installing and running SAP NetWeaver for Linux in the AWS Cloud.

This guide is intended for SAP architects, SAP engineers, IT architects, and IT administrators who want to deploy SAP NetWeaver on AWS.

About this Guide

This guide is part of a content series that provides detailed information about hosting, configuring, and using SAP technologies in the AWS Cloud. For the other guides in the series, ranging from overviews to advanced topics, see the SAP on AWS Technical Documentation home page.

Overview

Amazon Web Services provides various services and tools for deploying SAP products on the AWS Cloud platform. This guide discusses the steps required to use the AWS Command Line Interface (AWS CLI) and Linux commands to set up and configure AWS resources such as Amazon Elastic Compute Cloud (Amazon EC2) instances, Amazon Elastic File System (Amazon EFS), and Amazon Elastic Block Store (Amazon EBS) volumes to install a new SAP instance.

The guide also explains how to configure the SLES or RHEL operating systems for new SAP NetWeaver installations. By the end of this document, you will have the AWS infrastructure ready to install an SAP NetWeaver instance.

Costs and Licenses

You are responsible for all costs related to your use of any AWS services while following this guide. Prices are subject to change. For full details, see the pricing pages for the AWS services that you intend to use.

You must already own a license for the SAP software and have access to download the SAP software from the SAP Software Download Center (requires access to SAP Support).

Prerequisites

Specialized Knowledge

Creating AWS resources for SAP NetWeaver installation is an advanced topic. This guide assumes familiarity with AWS services and resources such as Amazon Virtual Private Cloud (Amazon VPC), Amazon EC2, Amazon EBS, and security groups. The guide does not cover the setup of Amazon VPC, subnets, and security groups, or the installation of SAP NetWeaver. If you are new to AWS, see the following content to get familiar with AWS technologies:

- Getting started with AWS
- Amazon EC2
- Amazon EBS
- Amazon VPC
- AWS CLI
- Amazon EFS

AWS also provides a Quick Start reference deployment to fast-track your SAP NetWeaver deployment in the AWS Cloud. The Quick Start leverages the AWS programmable infrastructure to provision resources quickly and deploy SAP NetWeaver—usually in less than an hour, with minimal manual intervention.

Technical Requirements

Before you get started, complete the following preparatory tasks:

1. Make sure that you have the following information for your existing AWS resources. You will need this information when you run AWS CLI commands to create your Amazon EC2 and Amazon EBS resources.

Information	Description
Region ID	Region where you want to deploy your AWS resources.
Availability Zone	Availability Zone within your target region where you want to deploy your resources

Information	Description
	. (For a High Availability installation, you would need two Availability Zones.)
Placement group	Placement group you want to use to logically group your Amazon EC2 instances within an Availability Zone (recommended for scale out SAP implementations).
Amazon VPC ID	Amazon VPC where you want to deploy your Amazon EC2 instance for SAP installation.
Subnet ID	Subnet where you want to deploy your Amazon EC2 instance.
Linux AMI ID	Amazon Machine Image (AMI) that will be used to launch your Amazon EC2 instance. You can find the latest Linux AMIs in AWS Marketplace.
Key pair	Make sure that you have generated the key pair in your target region, and that you have access to the private key.
Security group ID	Name of the security group that you want to assign to your Amazon EC2 instance.
Access key ID	Access key for your AWS account that will be used with AWS CLI tools.
Secret access key	Secret key for your AWS account that will be used with AWS CLI tools.

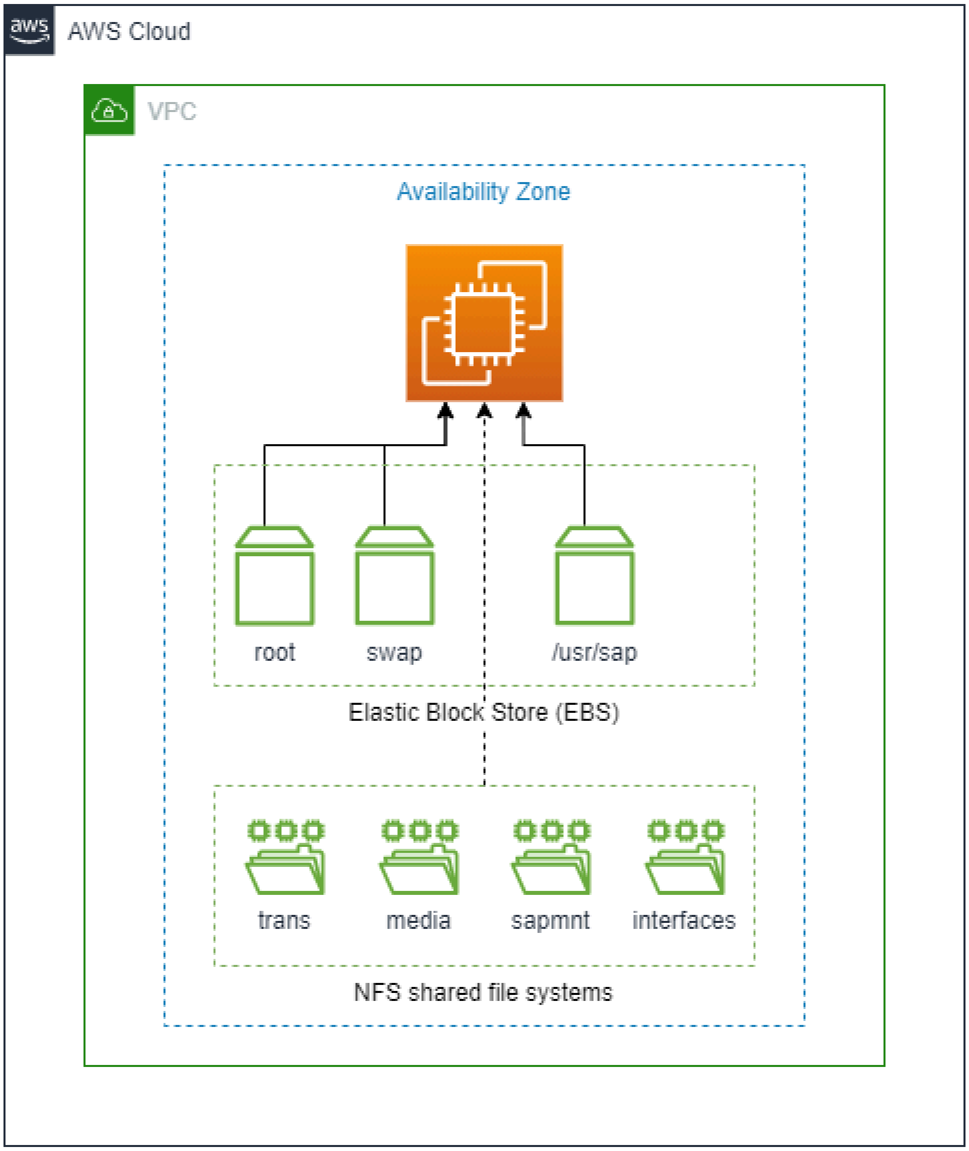
> **Note**
>
> If you are using SLES, we recommend installing SLES for SAP, which comes with a bundle of software that meets the specific needs for running SAP workloads on SLES. The SLES for SAP AMIs are available in AWS Marketplace.

2. Install the AWS CLI on the desktop, laptop, or server from which you plan to run the commands to deploy AWS resources. We recommend that you use the bundled installer. For instructions, see the AWS CLI User Guide.

> **Note**
>
> Configure your AWS CLI profile with the target region where you want to deploy your AWS resources for SAP installation.

Disk Layout

For SAP standard installation, we recommend the following file systems layout.



File system	Supported AWS storage service	Comments
root	Amazon Elastic Block Store (Amazon EBS)	N/A
swap	Amazon Elastic Block Store (Amazon EBS)	N/A
/usr/sap	Amazon Elastic Block Store (Amazon EBS)	N/A

File system	Supported AWS storage service	Comments
	Amazon Elastic File System (Amazon EFS)	
sapmnt	Amazon Elastic File System (Amazon EFS) Amazon FSx for NetApp ONTAP	Single Availability Zone or Multi-Availability Zone
trans	Amazon Elastic File System (Amazon EFS) Amazon FSx for NetApp ONTAP	Single Availability Zone or Multi-Availability Zone
interfaces	Amazon Elastic File System (Amazon EFS) Amazon FSx for NetApp ONTAP	Single Availability Zone or Multi-Availability Zone
media	Amazon Elastic File System (Amazon EFS) Amazon FSx for NetApp ONTAP	Single Availability Zone or Multi-Availability Zone

Note

In a standard installation, `/usr/sap` can also be mounted on Amazon EFS. Directories for interfaces and media are optional.

Creating AWS Resources and Configuring the Operating System for SAP NetWeaver Installation

In this scenario, we will provision an Amazon EC2 instance for installing SAP as a standard system; i.e., ABAP System Central Services (ASCS) and Primary Application Server (PAS) will run on one host.

> **Note**
>
> In this section, the syntax shown for the AWS CLI and Linux commands is specific to the scope of this document. Each command supports many additional options. For more information, use the AWS CLI `aws help` command or refer to the documentation.

Step 1. Check the Region Where You Want to Deploy Your AWS Resources

Display the AWS CLI configuration data.

```
$ aws configure list
```

In the command output, make sure that the default region that's listed is the same as the target region where you want to deploy your AWS resources and install SAP NetWeaver.

Step 2. Create a JSON File for the Amazon EBS Storage

Create a JSON file that contains the storage requirements for SAP Install volumes; for example:

```
[
 {
    "DeviceName": "/dev/sdh",
    "Ebs": {
      "VolumeSize";: 50,
      "VolumeType";: "gp2",
      "DeleteOnTermination": true
   }
 },
 {
    "DeviceName": "/dev/sdg",
    "Ebs": {
      "VolumeSize": 50,
      "VolumeType": "gp2",
      "DeleteOnTermination": true
   }
 }
]
```

Step 3. Launch the Amazon EC2 Instance

Launch the Amazon EC2 instance for the SAP installation in your target region by using the information that you gathered in the preparation phase. You will also be creating the required storage volumes and attaching them to the Amazon EC2 instance for the SAP installation, based on the JSON file that you created in the previous step.

Use the following syntax:

```
$ aws ec2 run-instances \
--image-id <AMI-ID> \
--count <number-of-EC2-instances> \
--instance-type <instance-type> \
--key-name=<name-of-key-pair> \
--security-group-ids <security-group-ID> \
--subnet-id <subnet-ID> \
--block-device-mappings file://C:\Users\file.json \
--region <region-ID>
```

The JSON file is the storage file that you created in step 2.

When using the command, make sure to place the command and its parameters on a single line; for example:

```
 $ aws ec2 run-instances --image-id ami-123456789abcdefgh --count 1 --instance-type
m5.large --key-name=my_key --security-group-ids sg-123456789abcdefgh --subnet-id
subnet-1234abcd --block-device-mappings file://C:\Users\storage.json
```

Step 4. Update the Hostname

Log in to your SAP Instance with Secure Shell (SSH) using the private key pair, and switch to root user to update the hostname along with the domain name according to your requirements. For detailed steps, see the AWS Knowledge Center article for your operating system:

- Amazon EC2 instance running on SLES
- Amazon EC2 instance running on RHEL

Alternatively, you can edit the `/etc/hosts` file and manually add this entry. For SAP systems, the hostname should not exceed 13 characters and should comply with SAP standards. See SAP OSS Note 611361 for details (requires access to SAP ONE Support Launchpad).

Step 5. Install Prerequisite Packages

Note

Your Amazon EC2 instance should have access to the internet to read and download required packages from the SUSE or Red Hat repository.

As root user, use the following commands to install the Linux packages that are required for SAP installation.

SLES syntax:

To install a package: `zypper install <package-name>`

To remove a package: `zypper remove <package-name>`

RHEL syntax:

To install a package: `yum install <package-name>`

To remove a package: `yum remove <package-name>`

1. Install `nfs-utils`, which is required for mounting the Amazon EFS mounts onto the Linux host.
2. Install the `nvme-cli` package to view the NVMe device mapping of Amazon EBS volumes.
3. Install SSM Agent by following the instructions in the AWS Systems Manager user guide.
4. Install SAP data provider.

```
cd /tmp
wget https://s3.amazonaws.com/aws-data-provider/bin/aws-agent_install.sh
chmod ugo+x aws-agent_install.sh
sudo ./aws-agent_install.sh
```

Step 6. Identify the Amazon EBS Device from NVMe Block Devices

On Nitro-based instances, device names that are specified in the block device mapping (step 2) are renamed as `/dev/nvme[0-26]n`. Before you proceed with the next step, ensure that you are using the appropriate device name to create a file system. Refer to the AWS documentation to learn how to find the Amazon EBS device on Nitro-based instances.

Step 7. Format Block Devices for Mounting SAP File Systems

To view the list of volumes attached to your instance and their device names, run the `lsblk` command as root user. The command displays the list of devices that are attached to your instance.

```
lsblk
NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme1n1     259:0    0  50G  0 disk
nvme0n1     259:1    0  10G  0 disk
#nvme0n1p1  259:2    0  10G  0 part /
nvme2n1     259:3    0  50G  0 disk
```

Format the block device for `/usr/sap`, swap, and other file systems that are needed to install SAP. As root user, format the Amazon EBS volumes attached to your instance to store local SAP files. You need to create a label for the file system as well. This label will be used to mount the file system.

```
mkfs.xfs -f /dev/nvme1n1 -L USR_SAP
```

Tip

NVMe device IDs associated with the volume could change during reboots. To avoid mount errors during instance reboots, you need to create a label for your file systems and mount them by label rather than by the actual NVMe IDs. This will also help in the situation where you need to change your instance type between Nitro-based and non Nitro-based instances.

Step 8. Create Directories and Mount the File System

As root user, create the directories to mount the file systems required for SAP installation. Start with the `/usr/sap` mount, using the syntax `mkdir <directory-path>`.

```
mkdir /usr/sap
```

As root user, add entries to the `/etc/fstab` file and mount the file systems. Adding entries to `/etc/fstab` ensures that your file systems are mounted automatically when your Amazon EC2 instance is restarted.

Add the entries for local SAP file systems to the `/etc/fstab` file by using the following commands:

```
echo "/dev/disk/by-label/USR_SAP /usr/sap xfs noatime,nodiratime,logbsize=256k 0 0"
>> /etc/fstab
```

To mount the file system that has been added to `/etc/fstab`, use the syntax `mount -a`.

```
mount -a
df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.8G  8.0K  3.8G   1% /dev
tmpfs           3.8G     0  3.8G   0% /dev/shm
tmpfs           3.8G  9.5M  3.8G   1% /run
/dev/nvme0n1p1  9.8G  1.4G  7.9G  15% /
tmpfs           3.8G     0  3.8G   0% /sys/fs/cgroup
tmpfs           769M     0  769M   0% /run/user/1000
/dev/nvme1n1     50G   33M   50G   1% /usr/sap
```

Here you can see that `/usr/sap` is mounted on device `/dev/nvme1n1`.

Step 9. Create Swap for SAP Installation

Linux swap functionality can improve the overall performance of the system and is a mandatory prerequisite for SAP installation. To determine the value for swap, follow the recommendations in the SAP Note 1597355.

To allocate swap on device `/dev/nvme2n1`, use the following commands:

```
lsblk
NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme1n1     259:0    0  50G  0 disk /usr/sap
nvme0n1     259:1    0  10G  0 disk
```

```
#nvme0n1p1  259:2    0  10G  0 part /
nvme2n1     259:3    0  50G  0 disk

mkswap -f /dev/nvme2n1 -L SWAP
Setting up swapspace version 1, size = 50 GiB (53687087104 bytes)
LABEL=SWAP, UUID=07291579-afb6-4e5f-8828-4c1441841f9b
swapon -L SWAP
swapon -s
Filename        Type        Size      Used   Priority
/dev/nvme2n1    partition  52428796  0      -1
```

Device `/dev/nvme2n1` is now allocated to be used as swap by the SAP application that will be installed on this host.

Step 10. Creating the Amazon EFS Mount for /usr/sap/trans and /sapmnt

To create an Amazon EFS file system and mount it on the Amazon EC2 instance, do the following:

1. Create a security group for Amazon EFS.

   ```
    $ aws ec2 create-security-group --group-name efs-sap-sg --description "Amazon EFS
   for SAP, SG for EFS " --vpc-id vpc-123456789abcdefgh
   ```

   Make a note of the security group ID that is displayed in the output.

   ```
   {
       "GroupId": "sg-abc12def "
   }
   ```

   In the example, the security group ID is `sg-abc12def`.

2. Create an inbound rule for the security group.

   ```
    $ aws ec2 authorize-security-group-ingress --group-id sg-abc12def --protocol tcp --
   port 2049 --cidr 0.0.0.0/0
   ```

3. Create an Amazon EFS file system.

   ```
   $ aws efs create-file-system --creation-token efsforsap
   ```

The command should display the following output:

```
  {
    "SizeInBytes": {
        "Value": 0
    },
    "CreationToken": "efsforsap",
    "Encrypted": false,
    "CreationTime": 1523374253.0,
    "PerformanceMode": "generalPurpose",
    "FileSystemId": "fs-abc12def",
    "NumberOfMountTargets": 0,
    "LifeCycleState": "creating",
    "OwnerId": "xxxxxxxxxxxx"
  }
```

Make a note of the `FileSystemId` value. In this example, `FileSystemId` is `fs-abc12def`.

4. Create the tag for `FileSystemId`.

```
  $ aws efs create-tags --file-system-id <FileSystemId>
 Key=<Name>,Value=<SomeExampleNameValue>
```

In this example, the key is `usrsap` and the value is ECC.

```
  $ aws efs create-tags --file-system-id fs-abc12def --tags Key=usrsap,Value=ECC
```

5. Create the mount target.

```
  $ aws efs create-mount-target --file-system-id fs-abc12def --subnet-  id subnet-
a98c8386 --security-group sg-abc12def
```

The command should display the following output:

```
  {
    "MountTargetId": "fsmt-123abc45",
    "NetworkInterfaceId": "xxxxxxxxxx",
    "FileSystemId";: "fs-abc12def ",
    "LifeCycleState": "creating",
    "SubnetId": "xxxxxxxxxxx",
    "OwnerId": "xxxxxxxxxxxx",
    "IpAddress": "x.x.x.x"
  }
```

Make a note of the `LifeCycleState` value, which is `creating` in the example.

6. Wait for a few minutes, and then check the status of creation by using the following command:

```
  $ aws efs describe-mount-targets --file-system-id fs-abc12def
```

The mount target `fsmt-061ab24e` is now available.

```
  {
    "MountTargets": [
        {
            "MountTargetId": "fsmt-061ab24e",
            "NetworkInterfaceId": " xxxxxxxxxx ",
            "FileSystemId": "fs-abc12def",
            ";LifeCycleState": "available",
            "SubnetId": " xxxxxxxxxxx ",
            "OwnerId": " xxxxxxxxxxx",
            "IpAddress": "x.x.x.x"
        }
     ]
  }
```

7. The DNS name for your file system on Amazon EFS should use the following naming convention:

```
<file-system-id.efs.aws-region>.amazonaws.com
```

In this example, `us-east-1` is the AWS Region.

```
fs-abc12def.efs.us-east-1.amazonaws.com
```

8. Use SSH to connect to your Amazon EC2 instance and create the mount point.

```
mkdir /usr/sap/trans
```

9. Mount the Amazon EFS file system. Mounting Amazon EFS using a DNS name is by default deactivated on a VPC. Instead, you can use an IP address to the corresponding file system.

```
sudo mount -t nfs -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2
 ip-address:/trans </usr/sap/trans>
```

This IP address can be found in step 7; for example:

```
sudo mount -t nfs -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2
 fs-abc12def.efs.us-east-1.amazonaws.com:/trans /usr/sap/trans
```

Repeat steps 8-9 to create other Amazon EFS mount points for `/sapmnt` and any other SAP software that needs to be shared between the Amazon EC2 instances.

Step 11. Installing SAP on Amazon EC2

Amazon EC2 is ready for SAP NetWeaver installation. Download the software from the SAP Support Portal and proceed with the installation.

> **Note**
>
> SAP Software Provisioning Manager (SWPM) Prerequisites Checker validates your installation host for compliance with most of the prerequisites defined in the installation guide. We recommend using this feature for your installations.

Additional Resources

AWS for SAP

The SAP NetWeaver Quick Start deployment guide provides detailed instructions for deploying SAP NetWeaver on the AWS Cloud by using AWS CloudFormation templates.

Refer to the AWS for SAP blog for architecture recommendations and best practices.

Security group for SAP application

The security group of the Amazon EC2 instance will need to be updated to allow the instance to communicate with databases and other SAP applications in the ecosystem. For information about managing security groups, refer to the Amazon EC2 Security Groups for Linux Instances documentation.

SAP documentation

Full list of SAP services and ports

SAP notes that are relevant to AWS:

- SAPS Ratings of AWS Instance types supported for SAP Note 1656099
- 1588667 - SAP on AWS: Overview of related SAP Notes and Web-Links
- 1588896 - Linux: Support Statement for SLES on Amazon Web Services
- 1618572 - Linux: Support Statement for RHEL on Amazon Web Services
- 2369910 - SAP Software on Linux: General information
- NetWeaver Installation Guides
- 1827960 - Adjusting operating system limits for SAP instances

Document Revisions

Date	Change	In sections
December 2018	Initial publication	—

SAP NetWeaver on AWS Deployment and Operations Guide for Windows

SAP specialists, Amazon Web Services

Last updated: November 2022

This guide provides guidance on how to set up AWS resources and the Microsoft Windows Server operating system to deploy SAP NetWeaver on Amazon EC2 instances.

This guide is intended for SAP architects, SAP engineers, IT architects, and IT administrators who want to deploy SAP NetWeaver on AWS.

About this Guide

This guide is part of a content series that provides detailed information about hosting, configuring, and using SAP technologies in the AWS Cloud. For the other guides in the series, ranging from overviews to advanced topics, see the SAP on AWS Technical Documentation home page.

This guide is for users who are responsible for planning, architecting, and deploying SAP NetWeaver on AWS. You should have a good understanding of AWS services, general networking concepts, Windows Server operating systems, and SAP NetWeaver administration. This document guides you through the steps required to successfully launch and configure the resources required for SAP NetWeaver on Windows.

Instructions in this document are based on the recommendations provided by SAP and Microsoft for SAP NetWeaver on Windows as described in the following OSS notes:

SAP NetWeaver on Windows OSS Notes

SAP OSS Note	Description
1656099	SAP Applications on AWS: Supported DB/OS and Amazon EC2 products
1409608	Virtualization on Windows
1732161	SAP Systems on Windows Server 2012 (R2)

SAP OSS Note	Description
2384179	SAP Systems on Windows Server 2016
2751450	SAP Systems on Windows Server 2019
1564275	Install SAP Systems Using Virtual Host Names on Windows
3143497	SAP Systems on Windows Server 2022

In addition, this document also follows best practices from AWS, Microsoft, and SAP for SAP NetWeaver deployments on Windows. See the recommended reading section for more details.

This document doesn't provide guidance on how to set up network and security constructs, such as Amazon Virtual Private Cloud (Amazon VPC), subnets, route tables, ACLs, NAT Gateway, AWS Identity and Access Management (IAM) roles, and AWS Security Groups. Instead, it focuses on how to configure and maintain the compute, storage, and operating system constructs for SAP NetWeaver deployment and operation on Windows on AWS.

SAP NetWeaver is also available to deploy on Linux. If you're considering using Linux, see the SAP NetWeaver Quick Start for Linux.

Prerequisites

Specialized Knowledge

Before you follow the configuration instructions in this guide, we recommend that you become familiar with the following AWS services. (If you are new to AWS, start with the Getting Started Resource Center.)

- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Virtual Private Cloud (Amazon VPC)
- AWS Identity and Access Management (IAM)
- Amazon Elastic Block Store (Amazon EBS)
- Amazon FSx
- Amazon Simple Storage Service (Amazon S3)

- AWS Systems Manager
- AWS CloudFormation
- AWS CloudTrail
- AWS Control Tower

Recommended Reading

We also recommend reading these overview and best practice guides:

- SAP on AWS Overview and Planning
- Getting Started with Architecting SAP on the AWS Cloud
- Best Practices for Windows on Amazon EC2

Technical Requirements

1. Ensure that any service limits are high enough and the current usage low enough to be able to launch the resources that you need. If necessary, request a service limit increase for the AWS resource that you're planning to use. In particular:
   a. Ensure that your EC2 service limits are sufficient to launch the instances that you need for your SAP NetWeaver system.
   b. Ensure that your VPC service limits are sufficient to launch a new VPC (if necessary) or individual network resources within your VPC, such as Elastic IP addresses.
2. Gather the following information about your existing AWS resources. You will need this information to create your Amazon EC2 and Amazon EBS resources using the AWS Command Line Interface (AWS CLI) commands:

AWS Resource Information Required

Information Needed	Description
Region ID	Region where you want to deploy your AWS resources
Availability Zone	Availability Zone within your target Region where you want to deploy your resources

Information Needed	Description
Amazon VPC ID	Amazon VPC where you want to deploy your Amazon EC2 instance for SAP installation
Subnet ID	Subnet where you want to deploy your Amazon EC2 instance
AMI ID	Amazon Machine Image (AMI) that will be used to launch your Amazon EC2 instance. You can find the latest Linux AMIs in AWS Marketplace
Key Pair	Make sure that you have generated the key pair in your target Region, and that you have access to the private key
Security Group ID	Name of the security group that you want to assign to your Amazon EC2 instance. See the appendix for detailed information about the security group for SAP instances
Access Key ID	Access key for your AWS account that will be used with AWS CLI tools
Secret Access Key	Secret key for your AWS account that will be used with AWS CLI tools

a. Ensure that you have a key pair that you can use to launch your Amazon EC2 instances. To import or create a new key pair, see Amazon EC2 Key Pairs and Windows Instances.

b. Ensure that you know the network details, such as VPC-ID and Subnet-ID, of the VPC where you plan to launch your Amazon EC2 instances to host your SAP NetWeaver application.

c. Ensure that you have the required ports open on the security group attached to your Amazon EC2 instance hosting your database, to allow communication between your database and your SAP NetWeaver application. If needed, create new security groups that allow network traffic over both the database ports and the SAP NetWeaver application ports. For a list of SAP ports, see TCP/IP Ports of All SAP Products.

3. If you plan to use the AWS Command Line Interface (AWS CLI) to launch your instances, ensure that you have installed and configured the AWS CLI with the appropriate credentials. See Configuring the AWS CLI for more details.
4. If you plan to use the AWS Management Console to launch your instances, ensure that your IAM user has permission to launch and configure Amazon EC2, Amazon EBS, etc. See the IAM User Guide for more details.
5. Ensure that you have the required SAP software available either via an S3 bucket or on a file share accessible from Windows, such as Amazon FSx. For the fastest installation experience, we recommend copying the required software to an EBS volume attached to the relevant EC2 instance before running the install. This is best set up as a separate volume (mapped to a new drive in Windows) that, after completion of the installation, can then be detached and either deleted or re-attached to other EC2 instances for further installations. We recommend using the AWS CLI for this. Be sure to assign the appropriate IAM role permissions to the EC2 instance to allow S3 access.
6. If the installation type is distributed or high availability (HA), it will need to be a domain-based installation and a domain controller is required. If desired, you can use AWS Directory Service for this purpose. AWS Directory Service for Microsoft Active Directory, also known as AWS Managed Microsoft AD, enables your directory-aware workloads and AWS resources to use managed Active Directory in AWS. For details, see AWS Directory Service and Create Your AWS Managed Microsoft AD directory.

   When doing a domain-based installation, `sapinst.exe` should be run by a user with domain administration privileges (but not the `<SID>adm` user) or a domain administrator must complete the appropriate preparatory steps. For more details, consult the SAP NetWeaver installation guide for your version of SAP NetWeaver.
7. To create an Amazon FSx file system, you need the following prerequisites:
   a. An AWS account with the permissions necessary to create an Amazon FSx file system and an Amazon EC2 instance. For more information, see Setting Up.
   b. An Amazon EC2 instance running Microsoft Windows Server in the VPC based on the Amazon VPC service that you want to associate with your Amazon FSx file system. For information on creating an EC2 Windows instance, see Getting Started with Amazon EC2 Windows Instances.
   c. Amazon FSx works with Microsoft Active Directory to perform user authentication. You join your Amazon FSx file system to an AWS Directory Service for Microsoft Active Directory. For more information, see Create Your File System.

d. This guide assumes that you haven't changed the rules on the default security group for your VPC. If you have changed them, you need to ensure that you add the necessary rules to allow network traffic from your Amazon EC2 instance to your Amazon FSx file system. For more details, see Security.

e. Install and configure the AWS Command Line Interface (AWS CLI).

For additional details on these prerequisites, see Prerequisites for Getting Started.

Planning the Deployment

Plan your SAP system landscape according to the SAP Master Guide for your version of SAP NetWeaver and your combination of operating system and database.

Topics

Select the Region

In choosing the Region for deployment, you'll need to consider some key factors. For more details, see our Overview and Planning guide.

- Service availability
  - Not all AWS services or features are available in all Regions. Verify that all services and features that you want to use in your deployment are available in the Region you choose. You can check availability on our website. If certain services or features are not available in your desired Region, there are alternatives that we mention in the guide.
  - For SAP workloads discussed in this guide, this is particularly true for:

  - EC2 instance types
  - Amazon FSx for Windows File Server
  - AWS Backup
- Proximity and connectivity options
- Data residency
  - You retain complete control and ownership over your data in the Region in which it is physically located, making it easy to meet regional compliance and data residency requirements.

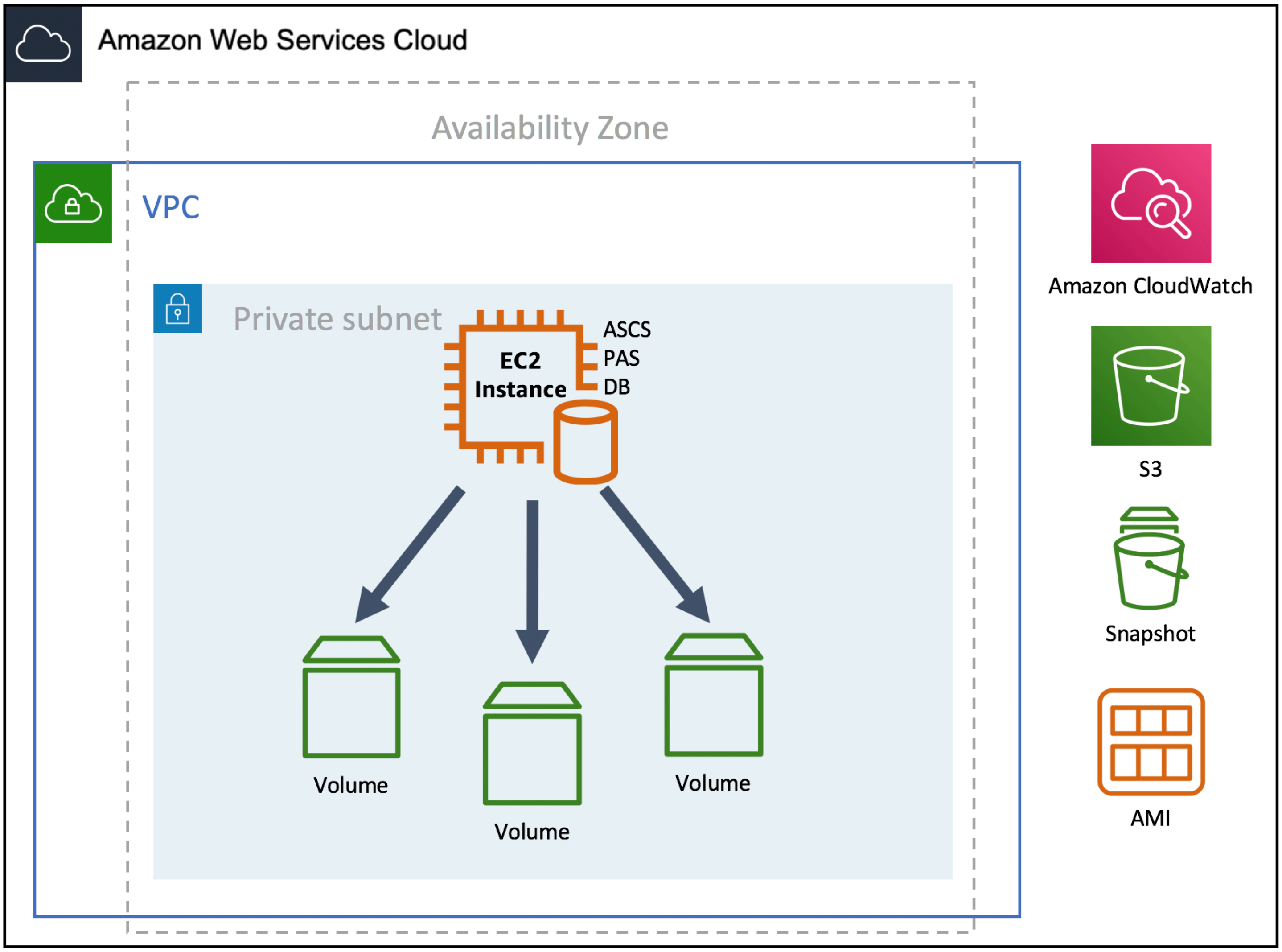
Architecture Options

Topics

- Standard System Deployment
- Distributed System Deployment
- High Availability System Deployment

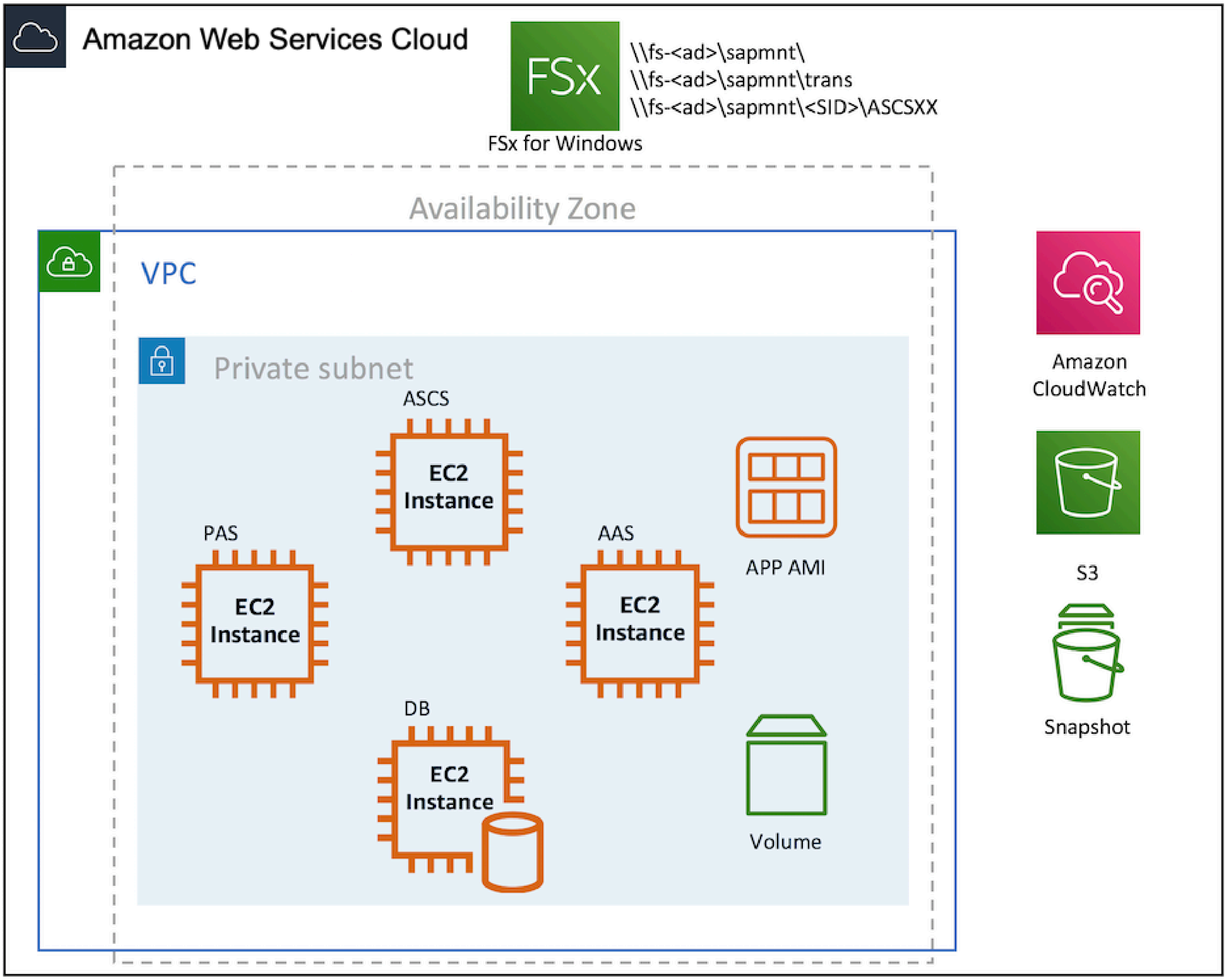
Standard System Deployment

Standard system or single host installation: all main instances of SAP NetWeaver (ASCS/SCS, database, and PAS) run on one Amazon EC2 instance. This option is best suited for non-production workloads.



Distributed System Deployment

Distributed system: every instance of SAP NetWeaver (ASCS/SCS, database, PAS, and optionally AAS) can run on a separate Amazon EC2 instance. This option is suited for both production and non-production workloads.



High Availability System Deployment

High availability (HA) system: used for business-critical applications. With this option, all the services that are single points of failure are deployed across multiple Availability Zones for fault tolerance.

For SAP NetWeaver, the key single points of failure are:

- the central services (ASCS/SCS)
- the global and transport filesystems

To protect against hardware failure of Amazon EC2 within an Availability Zone, you can enable EC2 instance recovery. See Recover Your Instance for more details on this feature. You can use scripts to start the SAP NetWeaver application automatically after instance recovery. You can further configure SAP application work processes to reconnect to your database after recovery. Consult the documentation for further restrictions. This option is not application aware and does not protect the application against Availability Zone failure, which makes it a good option for non-production systems. It also can be used for production systems but you might want to consider a Multi-AZ solution for this situation as well.

For HA solutions, it's important to be aware of two concepts within a VPC: shared storage and the Overlay IP address.

Shared Storage

EBS volumes are specific to a single Availability Zone and can only be attached to a single EC2 instance at a time. However, in distributed or HA deployments, shared storage is required for the global and transport filesystems. On AWS, this storage can be provided by building an NFS server or by using Amazon FSx. Amazon FSx provides shared file storage with full support for the SMB protocol, Windows NTFS, Active Directory integration, and Distributed File System (DFS).

If using such a solution in the context of a high availability installation, the shared storage solution you choose could introduce a single point of failure without appropriate protection. This can be protected against by:

- Clustering the NFS server providing the shared filesystem
- Clustering the host that is sharing the filesystems
- Using Amazon FSx. For workloads that require Multi-AZ redundancy to tolerate temporary AZ unavailability, you can create multiple file systems in separate AZs. Amazon FSx supports Microsoft's Distributed File System (DFS) Replication and Namespaces. DFS Replication allows you to automatically replicate data between two file systems, and DFS Namespaces allows you to configure automatic failover.

High availability

You can use a high availability (HA) clustering solution for autonomous failover of the central services across Availability Zones. There are multiple SAP-certified options for this clustering software on Windows listed on the SAP website, and it's also possible to build and automate your own solution. HA solutions that have been tested and are known to work on AWS include:

- Veritas InfoScale:
  - Veritas InfoScale for SAP on AWS
  - Veritas InfoScale for Windows compatibility list
- SIOS:
  - SIOS DataKeeper with Windows Server Failover Cluster (WSFC)
  - SIOS DataKeeper Cluster Edition on AWS Quick Start
  - SAP on AWS Blog: Implementing HA and DR for Microsoft SQL Server
- NEC ExpressCluster
- Windows Server Failover Cluster (WSFC) with native Windows and AWS services
  - SAP on AWS Blog: How to setup SAP NetWeaver on Windows MSCS for SAP ASCS/ERS on AWS

Support and certification

SAP clustering software is supported by the cluster software vendors themselves, not by SAP. SAP only certifies the solution. Any custom-built solution is **not** certified and will need to be supported by the solution builder.

In this guide, we focus on the distributed installation type on Windows in AWS. More details on how to deploy and operate SIOS, Veritas, and WSFC clusters are available on their respective websites linked above. For effective use of WSFC, Windows Server 2016, or later, is required.

The key features to be aware of with the WSFC solution are:

- ASCS and a separate ERS instance set up within Windows Cluster Manager
- Scale-Out File Server is a feature that is designed to provide scale-out file shares that are continuously available for file-based server application storage
- Storage Spaces Direct uses standard servers with local-attached drives to create highly available, highly scalable software-defined storage. **This requires a minimum of Windows Server 2016 and NVMe storage (so nitro-generation EC2 instances are required).**
- Amazon FSx for Windows File Server

Also read the High Availability with Microsoft Failover Clustering section of the SAP NetWeaver installation guide.

Security and Compliance

These additional AWS security resources can help you achieve the level of security that you require for your SAP NetWeaver environment on AWS:

- AWS Cloud Security Center
- CIS Amazon Web Services Foundations whitepaper
- Introduction to AWS Security
- AWS Security Best Practices whitepaper
- AWS Well-Architected Framework Security Pillar whitepaper
- Network and Security topic from the *Amazon EC2 User Guide for Windows Instances*

OS Hardening

You may want to lock down the OS configuration further, for example, to avoid providing a NetWeaver administrator with root credentials when logging into an instance.

We provide guidance on how to best secure your Windows EC2 instances:

- Read our best practices guide for securing Windows on EC2.
- Read our general best practices guide for securing EC2 instances.
- Use Amazon Inspector, an automated security assessment service that helps you test the network accessibility of your EC2 instances and the security state of your applications running on the instances.

You can also refer to the following SAP note:

- 1837765: Security policies for <SID>adm and SapService<SID> on Windows

Encryption

Cloud security at AWS is the highest priority. A core aspect of securing your workloads is encrypting your data—both at rest and in transit.

When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volume
- All volumes created from those snapshots

Encryption operations occur on the servers that host EC2 instances, ensuring the security of both data at rest, and data in transit between an instance and its attached EBS storage. You can expect the same IOPS performance on encrypted volumes as on unencrypted volumes, with a minimal effect on latency. Encryption and decryption are handled transparently and require no additional action from you or your applications.

Similarly, all Amazon FSx file systems are encrypted at rest with keys that are managed using AWS Key Management Service (AWS KMS). Data is automatically encrypted before being written to the file system, and automatically decrypted as it is read. These processes are handled transparently by Amazon FSx, so that you don’t have to modify your applications.

For Amazon S3, you can protect data in transit by using SSL/TLS or client-side encryption, and protect data at rest by using either server-side or client-side encryption.

You can find more information about encryption from the specific service documentation:

- Encrypting Amazon FSx Data at Rest and Data in Transit
- Protecting Amazon S3 Data Using Encryption
- Amazon EBS Encryption

Security Groups / NACLs

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. Security groups act at the instance level, not the subnet level.

Customers often separate the SAP system into multiple subnets, with the database in a subnet separate from the application servers, and other components, such as a Web Dispatcher, in another subnet—possibly with external access.

If workloads are scaled horizontally, or high availability is necessary, you might consider including multiple, functionally similar, EC2 instances in the same security group. In this case, you’ll need to add a rule to your security groups.

If Microsoft Windows Server is used, some configuration changes may be necessary in the security groups, route tables, and network access control lists (ACLs). You can refer to the operating system product documentation or other sources, such as the Security Group Rules Reference in the Amazon EC2 documentation, for more information.

A network access control list (ACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets (they’re stateless firewalls at the subnet level). You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

For further information on network considerations for SAP workloads, see our SAP on AWS network documentation.

API Call Logging

AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you. The information recorded includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service.

With CloudTrail, you can get a history of AWS API calls for your account, including API calls made via the AWS Management Console, AWS SDKs, command line tools, and higher-level AWS services, such as AWS CloudFormation. The AWS API call history provided by CloudTrailenables security analysis, resource change tracking, and compliance auditing.

Notifications on Access

You can use Amazon Simple Notification Service (Amazon SNS) or third-party applications to send notifications about SSH logins to your email address or mobile phone number.

Sizing

One of the first points to consider is whether this deployment is a completely new project (greenfield) or a migration. Sizing then applies across three key areas: compute, storage, and network.

Compute

Understanding the compute requirement helps you select the best matching EC2 instance type from the available list of SAP-certified instances.

If this is a greenfield deployment, use the SAP Quick Sizer tool to calculate the SAP Application Performance Standard (SAPS) compute requirement and use that value to select the EC2 instance that is the closest match with the best cost. Also check that the EC2 instance you select provides sufficient EBS and overall network throughput to satisfy your application requirements.

For migrations, you can use a number of data sources to help choose the best instance size:

- Source system utilization and workload patterns (EarlyWatch alert reports, etc.)
- Source system specification: CPU, memory, storage size, throughput, IOPS, network
- Source system SAPS rating

Selecting EC2 Instance Type

It's important to consider storage and network performance as well as compute, to ensure the selection of the best EC2 instance type.

After the workload is running on AWS, you can use a process called right sizing to refine the size that you actually need. Right sizing is best thought of as an on-going process.

Storage

Deploying SAP NetWeaver on Windows on AWS requires a minimum amount of storage and storage layout as per the SAP NetWeaver documentation for Windows. See the SAP documentation for further details on minimum and recommended storage sizes and storage layout. The EBS volumes should be created to match these requirements.

Verify that the amount of storage is adequate to provide sufficient I/O performance, as the performance of a General Purpose SSD (gp2) volume is related to the overall volume size. To achieve higher throughput and IOPS performance, the striping of volumes is often considered but this is usually not necessary for the NetWeaver application layer.

Network

Network performance is often not explicitly stated as a requirement in SAP sizing, but you can check the network performance of each EC2 instance type to ensure that you are delivering the required performance.

Operating System

If you plan on using Windows other than via Amazon EC2 for Windows Server, then ensure that you have the appropriate licenses and tenancy type selected. For more details, refer to your licensing terms and conditions, and see our Windows on AWS webpage.

A base AMI is required to launch an Amazon EC2 instance. For SAP NetWeaver workloads on Windows, you need to run Windows Server 2012 R2, or later, because older versions are no longer supported by SAP. If you are using bring your own license (BYOL) instead of license-included for Windows Server, you will need to create your own AMI. See Microsoft Licensing on AWS.

Ensure that you have access to the appropriate Windows Server AMIs before proceeding.

As with any operating system, we recommend that you keep the OS up-to-date with the latest patches. You can also refer to the following SAP Notes:

- 2325651: Required Windows Patches for SAP Operations

Compute

AWS has certified multiple instance families of various sizes for running SAP NetWeaver workloads. For a complete list of the certified EC2 instance types, see Amazon EC2 Instance Types for SAP.

Select the appropriate EC2 instance type based on your CPU, memory, and SAPS requirements. AWS recommends that, when possible, you use the latest generation of your selected instance family that is SAP certified.

Storage

Refer to the sizing section for resources on SAP's standard recommendations. If no storage performance requirements are available, AWS recommends General Purpose SSD (gp2) as the default EBS volume type for SAP workloads.

In practice, application servers will have a minimum of two volumes, mapped to the `C:` and `D:` drives. The `C:` drive is the boot volume containing the OS, and the `D:` drive is used to host the SAP software. We recommend using an additional, temporary volume for SAP software downloads (typically mapped as the `E:` drive).

If the installation type is distributed or HA, fileshares for the global filesystem and transport directories will need to be used across all relevant EC2 instances. In this guide, we use the standard

Windows file sharing features to share these directories from the EC2 instance hosting the central services. The `sapinst.exe` installer creates these shares automatically if it is run as a user with appropriate permissions.

Customers can also use NFS-based solutions, such as Amazon FSx, third-party solutions available from the AWS Marketplace, or custom-built solutions. Choosing the correct NFS solution is beyond the scope of this guide. If you use such a solution as part of a high availability deployment, consider that the NFS solution could itself be a single point of failure without appropriate protection.

Network

Ensure that you have your network constructs set up to deploy resources related to SAP NetWeaver. If you haven't already set up network components, such as Amazon VPC, subnets, and route tables, you can use the AWS Quick Start for Modular and Scalable VPC Architecture to easily deploy scalable VPC architecture in minutes. See the deployment guide for more details, then set up your EC2 instances for the NetWeaver application server within this VPC.

You also will need to set up a secured network connection between the corporate data center and the VPC, along with the appropriate route table configuration, if this has not already been configured.

Deployment Steps

Topics

Step 1: Prepare your AWS Account

In this example, we step through setting up a sample environment for the installation, which includes a public subnet for RDP and SSH access via the internet. In this scenario, we are using the AWS Quick Start for Modular and Scalable VPC Architecture in a Single-AZ deployment to create the VPC, subnets, security groups, and IAM roles. This setup is just an example and you should follow your own network layout and ensure that you comply with your security standards. This could include:

- Using an AWS Quick Start that suits their requirements such as a Multi-AZ deployment of the AWS Quick Start for SAP HANA
- Using a landing zone solution, like AWS Control Tower
- Working with your cloud team (for example, a Cloud Center of Excellence or CCoE) to ensure adherence to existing standards

  1. Check the Region where you want to deploy your AWS resources:

     a. You'll have picked the Region you want to deploy in during your planning phase.

     b. Display the AWS CLI configuration data:

     ```
     $ aws configure list
     ```

     In the command output, make sure that the default Region that's listed is the same as the target Region where you want to deploy your AWS resources and install SAP NetWeaver.

  2. If this is a distributed or HA installation type:

     a. Create a new security group specifically for the EC2 instances running the NetWeaver application servers that allows traffic over the required ports for remote access from the public subnet, for example, RDP.

     b. Edit that security group to allow traffic over ports required for SAP NetWeaver based on your specific use-case. Specify the source as being the security group itself and ensure that this security group is attached to all EC2 instances that will run application servers.

     c. For distributed or HA installations, ensure that the security group attached to each application and central services server allows communication between them over the required ports. You can create a rule that references a security group as its own source, and allow traffic on the required ports for that rule.

  3. Create a JSON file for the Amazon EBS storage volumes (the volume sizes used are indicative only and should be customized based on your sizing requirements):

     ```
     [
        {
             "DeviceName": "xvdb",
             "Ebs": {
                 "VolumeSize": 50,
     ```

```
                "VolumeType": "gp2",
                "DeleteOnTermination": true
            }
        },
        {
            "DeviceName": "xvdc",
            "Ebs": {
                "VolumeSize": 50,
                "VolumeType": "gp2",
                "DeleteOnTermination": true
            }
        }
    ]
```

4. AWS Windows AMIs provide additional software that prepares an instance when it first boots up. This is either the EC2Config service (Windows AMIs prior to Windows Server 2016) or EC2Launch (Windows Server 2016, or later). After the devices have been mapped to drives, they are initialized and mounted. The root drive is initialized and mounted as `C:\`. By default, when an EBS volume is attached to a Windows instance, it can show up as any drive letter on the instance. You can change the settings to set the drive letters of the volumes per your specifications. For more information, see the device naming section for storage on Windows.
5. Install your selected database product. If this is a distributed or high availability deployment, install your selected database product in a separate EC2 instance dedicated to that purpose. Otherwise, install your database in the existing EC2 instance. For more details, see the AWS Documentation for your database.
6. Launch EC2 instances for the SAP installation in your target Region by using the information you gathered in the preparation phase. You will also be creating the storage volumes required for the SAP installation and attaching them to the Amazon EC2 instance for the SAP installation.

   Ensure that you enable detailed monitoring on each instance as this is required for SAP support. (The sample commands provided below enable this.)

   Make sure that you choose one of the Amazon EC2 Instance Types for SAP. Sample AWS CLI syntax is given below.

```
$ aws ec2 run-instances \
--image-id <AMI-ID> \
--monitoring Enabled=true \
--count <number-of-EC2-instances> \
--instance-type <instance-type> \
--key-name=<name-of-key-pair> \
--security-group-ids <security-group-ID> \
--subnet-id <subnet-ID> \
--block-device-mappings https://<bucket>.s3.amazonaws.com/<file>.json
```

Example

This example enables detailed monitoring (data is available in 1-minute periods for an additional cost) which is a support prerequisite for SAP workloads on Amazon EC2.

```
$ aws ec2 run-instances \
--image-id ami-012345678901234ab \
--monitoring Enabled=true \
--count 1 \
--instance-type m5.2xlarge \
--key-name=my_key \
--security-group-ids sg-01234567890abcdef \
--subnet-id subnet-0123456789abcdefg \
--block-device-mappings https://example.s3.amazonaws.com/file.json
```

Step 2: Prepare Each EC2 Instance for SAP Installation

1. Log into the newly created RDP host in the public subnet. We will call this **jumpbox** for easy reference. Do this by either using AWS Systems Manager Session Manager (for command line tasks), or by doing the following:

   a. Go to the AWS Management Console, select the EC2 instance **jumpbox**, and choose **Connect**. Download the RDP file from the pop-up that appears.

   b. Click **Get Password** and provide your private key to decrypt the password. This is the password for the local administrator on **jumpbox**.

   c. Open the RDP file in your preferred RDP program, and connect to **jumpbox**. Log in with user Administrator and the password that you just retrieved in step 1b.

   d. After you are logged in, go back to the AWS Management Console and repeat step 1a and step 1b, but specify the EC2 instance where you will install NetWeaver. We'll call this **nw-ascs** for reference. Copy the downloaded RDP file to **jumpbox**.

   e. While logged into **jumpbox**, open the RDP file for **nw-ascs** in your preferred RDP program.

2. Log in as a user with administrator privileges but not an existing `<SID>adm` user (as per SAP's requirements).
3. Install the AWS CLI tools or use the AWS Tools for PowerShell provided with the Windows AMI.
4. Install the Java Runtime Environment (JRE) version that is compatible with your SAP installation software.
5. Install the AWS Data Provider, following the instructions for Windows in the Installation and Operations Guide.
6. Install and configure AWS Systems Management Agent (SSM Agent).

Step 3: Create Amazon FSx Volumes

1. The global fileshare and transport directories need to be available across all your SAP system's EC2 instances. In this guide, we assume that you are using Amazon FSx for this purpose.
2. Be sure that you've satisfied the prerequisites in the Technical Requirements section of this document. You will need to have already deployed your EC2 instances in each of the Availability Zones where you will create Amazon FSx filesystems.
3. Follow the step-by-step instructions in the Getting Started with Amazon FSx documentation
4. For high availability deployments that require Multi-AZ redundancy to tolerate temporary AZ unavailability, follow the instructions to create multiple file systems in separate AZs.

Step 4: Prepare and Run the SAP Installation Prerequisites Check

1. Download the SAP installation media for SWPM (the latest appropriate version for your desired NetWeaver installation), your desired NetWeaver software version for Windows, the latest compatible SAP kernel, and any other required files (such as: the host agent, IGS, database client tools, SAP GUI, the SAPCAR archiving tool, and the SAP download manager) to an attached EBS volume as described in the prerequisites (usually from Amazon S3 using the AWS CLI tools).
2. Run the SAP prerequisite checker via SWPM on the desired host servers to ensure that you have met SAP's technical prerequisites. When you first run SWPM, you may have to enter the sign-in credentials of the Windows user that you're currently logged in as.
3. Launch SWPM by running the `sapinst.exe` executable. Specify `SAPINST_USE_HOSTNAME=<FQDN>` when launching to override the default DNS name if necessary, for example, with `<hostname>.local`.

4. Complete the recommended prerequisite steps as identified by the SAP prerequisite checker as per your specific requirements. Some common prerequisites for Windows Server operating systems are:

   a. Ensure that the hostname is ⇐ 13 characters in an alphanumeric string (hyphens can also be included). This can be done at the command line using Windows PowerShell by executing the following command:

      ```
      Rename-Computer <new-hostname>
      ```

   b. Optionally add the server to your Active Directory domain (this can be done with AWS Systems Manager).

   c. Pagefile size will have a minimum recommended value based on services selected.

   d. Continuous Availability feature on Windows Server 2012 R2 can result in long wait times. See SAP note 1823833 for a fix.

Step 5: Install SAP NetWeaver on Amazon EC2

You are now ready to install SAP NetWeaver on this EC2 instance using the downloaded software. Proceed with the instructions in the SAP installation guide for your version of SAP NetWeaver.

You will need to do this for a minimum of:

- the ASCS instance
- the DB instance (on the installed database server)
- the PAS instance

and optionally for:

- other AAS instances
- ERS instance on the second ASCS node (in different AZ)

Operations

Topics

- Tagging AWS Resources

- Monitoring
- Backup and Restore
- Storage
- Operating System Maintenance
- High Availability
- Disaster Recovery
- Compute
- Cost Optimization
- Automation
- Support

Tagging AWS Resources

A tag is a label that you assign to an AWS resource. Each tag consists of a *key* and an optional *value*, both of which you define. Adding tags to the various AWS resources will not only make managing your SAP environment much easier but can also be used to quickly search for resources. Many Amazon EC2 API calls can be used with a special tag filter. Refer to AWS Tagging Strategies and use it as a starting point to define the tags you need for your resources. Some examples on how you can use tags for operational needs are:

- You can tag your EBS Volumes to identify their environment (for example Environment= DEV/ QAS/PRD etc.) and use these tags to create backup policies for EBS Volumes
- You can use similar tags as in above example with EC2 instances and use them for patching your operating systems or running scripts to stop/start application or EC2 instances.

Monitoring

AWS provides multiple native services to monitor and manage your SAP environment. Services like CloudWatch and CloudTrail can be leveraged to monitor your underlying infrastructure and APIs respectively. CloudWatch provides ready-to-use KPIs for CPU, disk utilization and also allows you to create custom metrics if your specific KPIs that you would like to monitor. CloudTrail allows you to log the API calls made to your AWS infrastructure components.

Backup and Restore

Snapshots and AMIs

A common approach for backing up your SAP NetWeaver application servers is using snapshots and AMIs.

All your data is stored on Amazon EBS volumes attached to the SAP NetWeaver application servers. You can back up the data on these volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups of Amazon EBS volumes, which means that only the blocks on the device that have changed after your most recent snapshot are saved. For more details on this, see Creating an Amazon EBS Snapshot.

An Amazon Machine Image (AMI) provides the information required to launch an instance along with a block device mapping of all EBS volumes attached to it.

Amazon EC2 powers down the instance before creating the AMI to ensure that everything on the instance is stopped and in a consistent state during the creation process. If you're confident that your instance is in a consistent state appropriate for AMI creation, you can check the No Reboot option.

To take application-consistent snapshots of all EBS volumes attached to your instance using Windows Volume Shadow Copy Service (VSS), see Creating a VSS Application-Consistent Snapshot. This allows you to create a copy of the image without rebooting the instance.

You can use AWS Backup to centrally configure backup policies and monitor backup activity for these snapshots.

After you have completed the SAP installation and post installation steps, you should create an image of the instance. AWS provides a very simple and quick way to copy an SAP system. You can use the AWS Management Console or the AWS CLI to create a new AMI of an existing SAP system. The new AMI contains a complete copy of the operating system and its configuration, software configurations, and all EBS volumes that are attached to the instance. From the new AMI, you can launch exact copies of the original system. For details on how to create an AMI of an existing EC2 instance, see Creating a Custom Windows AMI.

Example:

```
 $ aws ec2 create-image --instance-id i-1234567890abcdef0
--name "My server" --description "An AMI for my server"
```

Note

When you build an instance using an AMI, make sure that you update the hostname and the `C:\Windows\System32\Drivers\etc\hosts` file with the new metadata. These details usually get copied from the source.

File Backup to Amazon S3

You can perform traditional file-based backups from your EBS volumes to Amazon S3. One way to do this is by using the AWS CLI and trigger this using AWS Systems Manager Run Command so that you can centrally manage these.

Third-party Options

There are many third-party backup products for AWS services, including many solutions that have been certified by SAP. For more information, see AWS SAP Partner Solutions.

Amazon FSx Backup

With Amazon FSx, backups are file-system-consistent, highly durable, and incremental. To ensure file system consistency, Amazon FSx uses the Volume Shadow Copy Service (VSS) in Microsoft Windows. To ensure high durability, Amazon FSx stores backups in Amazon S3. Amazon FSx backups are incremental, which means that only the changes made after your most recent backup are saved.

Amazon FSx automatically takes backups of your file systems once a day. These daily backups are taken during the daily backup window that you established when you created the file system.

If you want to set up a custom backup schedule, you can deploy our reference solution.

Storage

The storage services we use across this guide are:

- Amazon EBS
  - Provides persistent storage for SAP application and database. The EBS volumes can be resized and even the EBS volume type can be changed without disrupting the applications. For more information, see Requesting Modifications to Your EBS Volumes. You will need to extend the filesystem to match the extended volume size using the Windows operating system tools.

- Amazon FSx for Windows File Server
  - Does not need you to explicitly provision storage at all – you simply pay for what you use.
  - Does need regular maintenance, but you can define your own maintenance window as per Amazon FSx Maintenance Windows.
  - The Amazon FSx Service Level Agreement provides for a service credit if your monthly uptime percentage is below our service commitment in any billing cycle.
- Amazon S3
  - Does not need you to explicitly provision storage at all – you simply pay for what you use.
  - You can use Object Lifecycle Management to set rules that define when objects are transitioned or archived to colder storage, such as S3 Standard-IA, S3 Glacier, or S3 Glacier Deep Archive, and when they expire. These actions happen automatically after being set.

Operating System Maintenance

In general, operating system maintenance across large numbers of EC2 instances can be managed by:

- Tools specific to each operating system, such as Microsoft System Center
- Third-party products, such as those available in AWS Marketplace
- Using AWS Systems Manager

Patching

You can follow SAP recommended patching processes to update your landscape on AWS. For operating system patching, with AWS Systems Manager Patch Manager you can roll out OS patches as per your corporate policies. There are multiple key features like:

- Scheduling based on tags
- Auto-approving patches with lists of approved and rejected patches
- Defining patch baselines

AWS Systems Manager Patch Manager integrates with IAM, AWS CloudTrail, and Amazon CloudWatch Events to provide a secure patching experience that includes event notifications and the ability to audit usage. For details about the process, see How Patch Manager Operations Work.

If AWS Systems Manager Patch Manager does not satisfy your requirements, there are third-party products available as well. Some of these products are available in the AWS Marketplace.

Maintenance Window

AWS Systems Manager Maintenance Windows lets you define a schedule for when to perform potentially disruptive actions on your instances, such as patching an operating system, updating drivers, installing software, or applying patches.

Administrator Access

You can access the backend SAP systems for administration purposes using:

- AWS Systems Manager Session Manager
- Remote Desktop Protocol (RDP)
- Secure Shell (SSH)

High Availability

After your HA cluster is deployed and configured successfully on AWS, the operation of the HA software still follows the third-party software interface. This can be best understood by following the operational guides from the respective vendors.

It's also important to have a test environment available (often called a staging or pre-production environment) that has an identical cluster configuration to your production environment. This environment can be used to test any configuration changes to the cluster before deploying the changes to production.

Two key AWS features that support the cluster software are:

- Amazon FSx for shared storage: See the storage section for maintenance considerations for Amazon FSx. For Multi-AZ deployments, DFS replication is required across multiple filesystems so ensure that you monitor the replication.
- Overlay IP for IP failover
  - Ensure that IAM authorizations are in place to minimize update access to the route table so that only the cluster agent can edit it.

- Ensure that the route table configuration is coupled with your change management process so that any wider environment updates that might affect this feature are captured and can therefore be tested.

Disaster Recovery

Network Fileshare Copy Out-of-Region

To ensure that you have an independent backup of your data in the secondary Region, you should back up your shared filesystem, as this won't be included in any AMIs or individual EBS snapshots.

To back up your Amazon FSx filesystem, you can rely on the included backup feature. However, this backs up to Amazon S3 in-region. To support out-of-region disaster recovery (DR), you will need to perform a file-level backup of your Amazon FSx filesystem in your secondary Region. You can do this by accessing the filesystem via cross-region VPC peering and then running a file-level copy from an Amazon EC2 instance running in the secondary Region to Amazon S3. This action can be automated and scheduled using AWS Systems Manager Run Command in combination with Amazon CloudWatch Events.

Fail-back Plan

When your primary Region returns to normal operations, you may consider failing back to it. In the event of a disaster that triggered a recovery to another Region, you copy the AMIs and shared filesystems from the secondary Region back to the primary. In other words, you'll reverse what was the regular process before the disaster. It's important to update your change management or change control processes to reflect this.

AMI Copy

When your primary AWS Region is affected by an availability event, AMIs allow you to quickly recover your SAP NetWeaver application servers in your DR Region. For recovery across Regions, ensure that the latest AMIs are copied to the disaster recovery Region using the AMI copy feature. New AMIs should be created when there are filesystem level changes to the SAP NetWeaver application servers. This can be caused by:

- SAP kernel changes
- Database client software updates
- Operating system patches

To ensure that you reliably create a new AMI when these events happen, add the AMI creation as a step in your change management process. It’s important that if using a mechanism like this that you integrate the out-of-region AMI copy with this process.

If having the lowest possible recovery time objective (RTO) is a priority, consider keeping at least one application server running in the secondary Region to minimize the recovery time.

AWS Elastic Disaster Recovery

AWS Elastic Disaster Recovery (Elastic Disaster Recovery) minimizes downtime and data loss with fast, reliable recovery of on-premises and cloud-based applications using affordable storage, minimal compute, and point-in-time recovery.

You can increase IT resilience when you use AWS Elastic Disaster Recovery to replicate on-premises or cloud-based applications running on supported operating systems. Use the AWS Management Console to configure replication and launch settings, monitor data replication, and launch instances for drills or recovery.

For more information, see the following resources.

- What is Elastic Disaster Recovery?
- Disaster recovery for SAP workloads on AWS using AWS Elastic Disaster Recovery

Compute

EBS volumes are exposed as NVMe block devices on Nitro-based instances. When changing EC2 instance types from a previous generation to a Nitro generation, and if using a Windows Server 2008 R2, or later, Windows AMI, the AWS NVMe driver is already included as described in the Amazon EBS and NVMe documentation. If you are not using the latest Windows AMIs provided by Amazon, see Installing or Upgrading AWS NVMe Drivers.

Besides operating system maintenance, there is also maintenance that you can consider for the EC2 instances themselves. This maintenance can be driven by AWS Systems Manager Automation documents. Some examples are:

- Use the AWS-StopEC2InstanceWithApproval document to request that one or more IAM users approve the instance stop action. After the approval is received, automation stops the instance.
- Use the AWS-StopEC2Instance document to automatically stop instances based on a schedule by using Amazon CloudWatch Events or a Maintenance Window task. For example, you can

configure an automation workflow to stop instances every Friday evening, and then restart them every Monday morning.

- Use the AWS-UpdateCloudFormationStackWithApproval document to update resources that were deployed using an AWS CloudFormation template. The update applies a new template. You can configure the automation to request approval by one or more IAM users before the update begins.

We also provide an AWS Solution called AWS Instance Scheduler that enables you to easily configure custom start and stop schedules for their Amazon EC2 and Amazon Relational Database Service (Amazon RDS) instances.

Cost Optimization

We recommend that you make cost optimization an on-going process. This is an extensive topic with many AWS services that help with budgeting, cost control, and proactive cost optimization recommendations.

For more details, see SAP on AWS Pricing and Optimization guide.

Automation

Automation using Infrastructure as Code with AWS CloudFormation

We recommend following the infrastructure as code principle in automating and maintaining your workloads on AWS. AWS CloudFormation provides a common language to describe and provision all the infrastructure resources in your cloud environment in a repeatable and automated manner.

Automation using Documents

AWS Systems Manager Automation simplifies common maintenance and deployment tasks associated with Amazon EC2 instances and other AWS resources. Automation enables you to do the following:

- Build Automation workflows to configure and manage instances and AWS resources.
- Create custom workflows or use pre-defined workflows maintained by AWS.
- Receive notifications about Automation tasks and workflows by using Amazon CloudWatch Events.

- Monitor Automation progress and execution details by using the Amazon EC2 or the AWS Systems Manager console.

There are many AWS-provided documents specific to Windows already available.

Support

To get help from SAP, SAP requires, at the minimum, a business support agreement with AWS. AWS Business Support provides resources and technical support for customers running SAP workloads on AWS. If you have any AWS-related technical issues, you can open a case with either SAP or AWS, and it will be routed to the appropriate teams. AWS also offers AWS Enterprise Support for customers running mission critical production workloads on AWS.

Additional Reading

SAP on AWS Technical Documentation

- SAP on AWS Technical Documentation
- SAP on AWS Whitepapers
- SAP NetWeaver on AWS Quick Start

  This is for SAP NetWeaver deployments on Linux, but is a useful point of comparison if you are looking to automate a Windows-based deployment, or implement a standard Multi-AZ network layout.
- AWS for SAP Blog
- Making Application Failover Seamless by Failing Over Your Private Virtual IP Across Availability Zones

SAP Documentation

- SAPS Ratings of AWS Instance types supported for SAP Note 1656099
- 1588667 - SAP on AWS: Overview of related SAP Notes and Web-Links
- 1656250 - SAP on AWS: Support prerequisites
- 2539944 - Windows Server / Microsoft SQL Server on AMI
- 1409604 - Virtualization on Windows: Enhanced monitoring

- 2198693 - Key Monitoring Metrics for SAP on Amazon Web Services

Document Revisions

Date	Change
November 2022	Added SAP Note 3143497 to SAP NetWeaver on Windows OSS Notes list
July 2019	Initial publication

Microsoft SQL Server for SAP NetWeaver on AWS Deployment and Operations Guide

SAP specialists, Amazon Web Services

Last updated: December 2020

This guide provides guidance on how to set up AWS resources and the Microsoft Windows Server operating system to deploy Microsoft SQL Server for SAP NetWeaver on Amazon EC2 instances.

This guide is for users who are responsible for planning, architecting, and deploying SQL Server on AWS for SAP NetWeaver based applications. You should have a good understanding of AWS services, general networking concepts, Windows Server operating systems, and SQL Server administration.

Overview

This guide is part of a content series that provides detailed information about hosting, configuring, and using SAP technologies in the Amazon Web Services Cloud. For the other guides in the series, ranging from overviews to advanced topics, see SAP on AWS Technical Documentation home page.

This guide provides guidance on how to set up AWS resources and the Microsoft Windows Server operating system to deploy Microsoft SQL Server for SAP NetWeaver on Amazon EC2 instances.

Instructions in this document are based on recommendations provided by SAP and Microsoft for SQL Server deployment on Windows via the below SAP notes or KB articles:

Table 1 - SAP NetWeaver on Windows OSS Notes

SAP OSS Note	Description
1656099	SAP Applications on AWS: Supported DB/OS and Amazon EC2 products
1409608	Virtualization on Windows
1732161	SAP Systems on Windows Server 2012 (R2)
2384179	SAP Systems on Windows Server 2016

SAP OSS Note	Description
2751450	SAP Systems on Windows Server 2019
1564275	Install SAP Systems Using Virtual Host Names on Windows
1772688	SQL Server AlwaysOn and SAP applications

In addition, this document also follows best practices from AWS, Microsoft, and SAP for SAP NetWeaver deployments on Windows.

This guide is for users who are responsible for planning, architecting, and deploying SQL Server on AWS for SAP NetWeaver based applications. You should have a good understanding of AWS services, general networking concepts, Windows Server operating systems, and SQL Server administration.

This document doesn’t provide guidance on how to set up network and security constructs like Amazon Virtual Private Cloud (Amazon VPC), subnets, route tables, ACLs, NAT Gateway, IAM Roles, AWS Security Groups, and so on. This document focuses on configuring and maintaining compute, storage, and operating system for Microsoft SQL Server for SAP NetWeaver based applications.

Prerequisites

Specialized Knowledge

Before you follow the instructions in this guide, we recommend that you become familiar with the following AWS services. (If you are new to AWS, see Getting Started with AWS.)

- Amazon EC2
- Amazon EBS
- Amazon FSx
- Amazon VPC
- AWS CloudFormation
- AWS Systems Manager
- Amazon Simple Storage Service (Amazon S3)
- AWS Identity and Access Management (IAM)

Technical Requirements

Before you start to deploy Microsoft SQL Server database for SAP applications on AWS, ensure that you meet the following requirements:

- Windows Server 2008 R2, 2012 R2, or 2016 operating system
- Microsoft SQL Server 2008 R2 or higher database
- Install AWS SAP Data provider on Amazon EC2 instances after installing SQL Server database
- If you plan to deploy domain installation, you should have a user ID that is a member of domain admins. Otherwise, the domain admin should create groups and user IDs (such as <sapsid>adm, SAPService<SAPSID>, and so on) as required for SAP in advance. See SAP installation guide for more details.
- AWS Account with permission to create resources.
- Access to SAP installation media for database and application
- AWS Business Support or AWS Enterprise Support plan

Planning

Topics

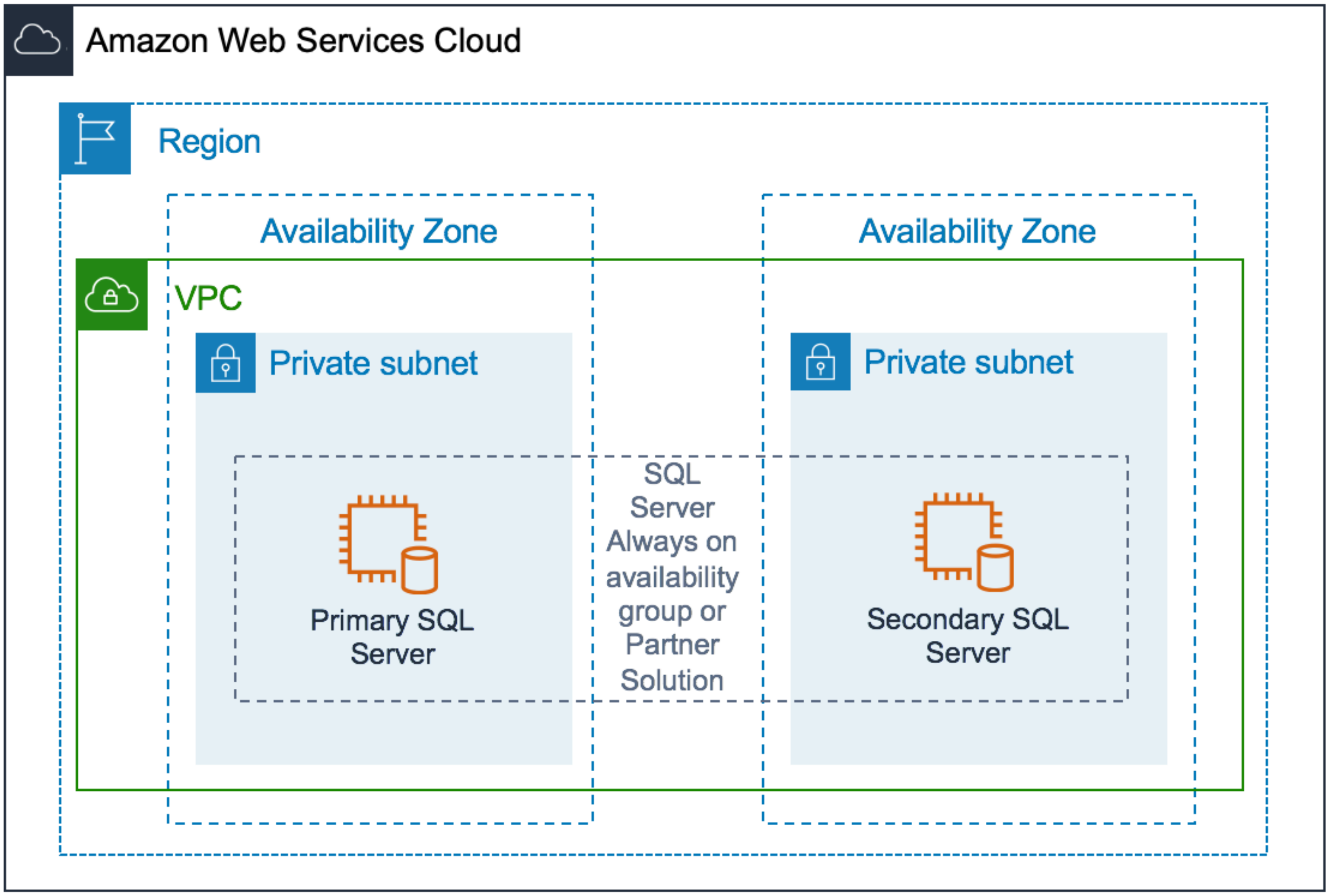
- Architecture Options
- Deployment Options
- Security
- Sizing
- Operating System
- Compute
- Storage
- Network
- Business Continuity

Architecture Options

SAP NetWeaver applications based on SQL Server can be installed in three different ways:

- **Standard system or single host installation**: ABAP System Central Services (ASCS)/System Central Services (SCS), Database, and Primary Application Server (PAS) of SAP NetWeaver run in single Amazon EC2 instance. This option is suited for non-critical and non-production workloads.
- **Distributed system**: ASCS/SCS, Database, and PAS of SAP NetWeaver run on separate Amazon EC2 instances. For example, you can choose to run ASCS and PAS on one Amazon EC2 instance and database on another Amazon EC2 instance or other possible combinations. This option is suited for production and non-production workloads.
- **High Availability (HA) system**: For your SAP application to be highly available, you need to protect the single point of failures. Database is one single point of failure in SAP applications. There are two methods you can use to protect SQL Server and make it highly available.
  - Database native solution: SQL Server Always On availability group.
  - Third-party solutions: For example, SIOS Data Keeper, NEC ExpressCluster, Veritas InfoScale.

Regardless of which option you choose to make your SQL Server database highly available, AWS recommends that you deploy a primary and secondary SQL Server in different AWS Availability Zones within an AWS Region. The following diagram provides a high-level architecture for SQL Server high availability on AWS. This option is suited for business-critical applications.



Deployment Options

Microsoft SQL Server 2008 R2 or later is supported for SAP applications on AWS. See SAP Note 1656099 - SAP Applications on AWS: Supported DB/OS and Amazon EC2 products for supported SAP applications and databases on AWS.

Security

AWS provides several security capabilities and services to securely run your SAP applications on AWS platform. In the context of SQL Server for SAP applications, you can use network services and features such as Amazon VPC, AWS Virtual Private Network, AWS Direct Connect, and Amazon EC2 security groups, network access controls, route tables, and so on, to restrict the access to your database.

Network Security

Generally, databases for SAP applications do not require direct user access. We recommend that you only allow network traffic to the Amazon EC2 instance running SQL Server from Amazon EC2 instances running SAP application servers (PAS/AAS) and ASCS/SCS.

By default, SQL Server receives communication on TCP port 1433. Depending on your VPC design, you should configure Amazon EC2 security groups, NACLs, and route tables to allow traffic to TCP Port 1433 from SAP application servers (PAS/AAS) and ASCS/SCS.

Encryption

We recommend that you encrypt your data stored in AWS storage services. See the following documentation for more details:

- Encrypting Data at Rest and in Transit for Amazon FSx
- Protecting S3 objects using encryption
- Amazon EBS Encryption

Sizing

SAP Quick Sizer is generally used to size the SAP environment for new implementations. However, if you are migrating your existing SAP applications based on SQL Server to AWS, consider using the following additional tools to right-size your SAP environment based on current use.

- **SAP Early Watch Alerts (EWA):** SAP EWA reports are provided by SAP regularly. These reports provide an overview of historical system use. Analyze these reports to see if your existing SAP system is overused or underused. You can use this information to right size your environment.
- **Windows native tools:** Gather and analyze historical use data for CPU/Memory with Performance Monitor/Windows System Resource Manager to right size your environment.
- **AWS Application Discovery Service:** AWS Application Discovery Service helps with collecting usage and configuration data about your on-premises servers. You can use this information to analyze and right-size your environment.

Since it is easy to scale up or scale down your Amazon EC2 instances on AWS, we recommend that you consider the following guidelines when sizing your SAP environment on AWS.

- Do not add too much capacity to meet future demand.

- Account for the SAP Quick Sizer buffer. SAP Quick Sizer tools provide sizing guidance based on assumptions that for 100% load (as per your inputs to tool) system use will not exceed 65%. Therefore, there is a fair amount of buffer already built into SAP Quick Sizer recommendation. See SAP’s Quick Sizer guidance for details.

Operating System

SAP applications based on SQL Server are supported only on Windows operating system. For supported Windows version, see the SAP product availability matrix (PAM) for the SAP application that you plan to deploy on AWS.

Compute

AWS provides multiple SAP certified Amazon EC2 instances. See SAP Note 1656099 - SAP Applications on AWS: Supported DB/OS and Amazon EC2 products for details. Based on results of your sizing exercise, you can deploy your SQL Server on any of the SAP certified Amazon EC2 instances that meets your requirement.

Storage

The following table lists the main directories for SQL Server database.

Table 2 - Main directories for SQL Server database

Usage	Directory	Description

Database data files	`<drive>:\<SAPSID>DATA0`
	`<drive>:\<SAPSID>DATA1`
	…..
	`<drive>:\<SAPSID>DATA<N>`
		Directory for SAP database data files

Database transaction log files	`<drive>:\<SAPSID>log<N>`	Directory for SAP database transaction Log

Usage	Directory	Description
Tempdb data files	`<drive>:\Tempdb`	Directory for temporary database data files
SQL binaries and other data files	`<drive>:\Program Files\Microsoft SQL Server`	Directory for SQL Server program files and master, msdb, and model data files

Amazon Elastic Block Store (Amazon EBS) volumes are designed to be highly available and reliable. Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component. Due to this built-in protection, you don't have to configure RAID 1 for volumes containing database transaction log files, tempdb data files, SQL binaries, and other data files.

We also do not recommend RAID 5 for database data files on AWS due to following reasons.

- Volumes are replicated within Availability Zone by default.
- Parity write operations of RAID 5 consume some of the IOPS available to your volume and will reduce the overall IO available for database operations by 20-30% over RAID 0 configuration.

Network

Ensure that your network constructs are set up to deploy resources related to SAP NetWeaver. If you haven't already set up network components like Amazon VPC, subnets, route tables, and so on, you can use AWS Quick Start for VPC to easily deploy scalable VPC architecture.

Business Continuity

We recommend that you architect your business-critical applications to be fault tolerant. Depending on your availability requirements, there are different ways in which you can achieve this. This section discusses how you can set up highly available SQL Server for SAP applications.

High Availability

You can configure high availability for SQL Server database on AWS using Always On availability groups or third-party tools.

SQL Server Always On Availability Groups

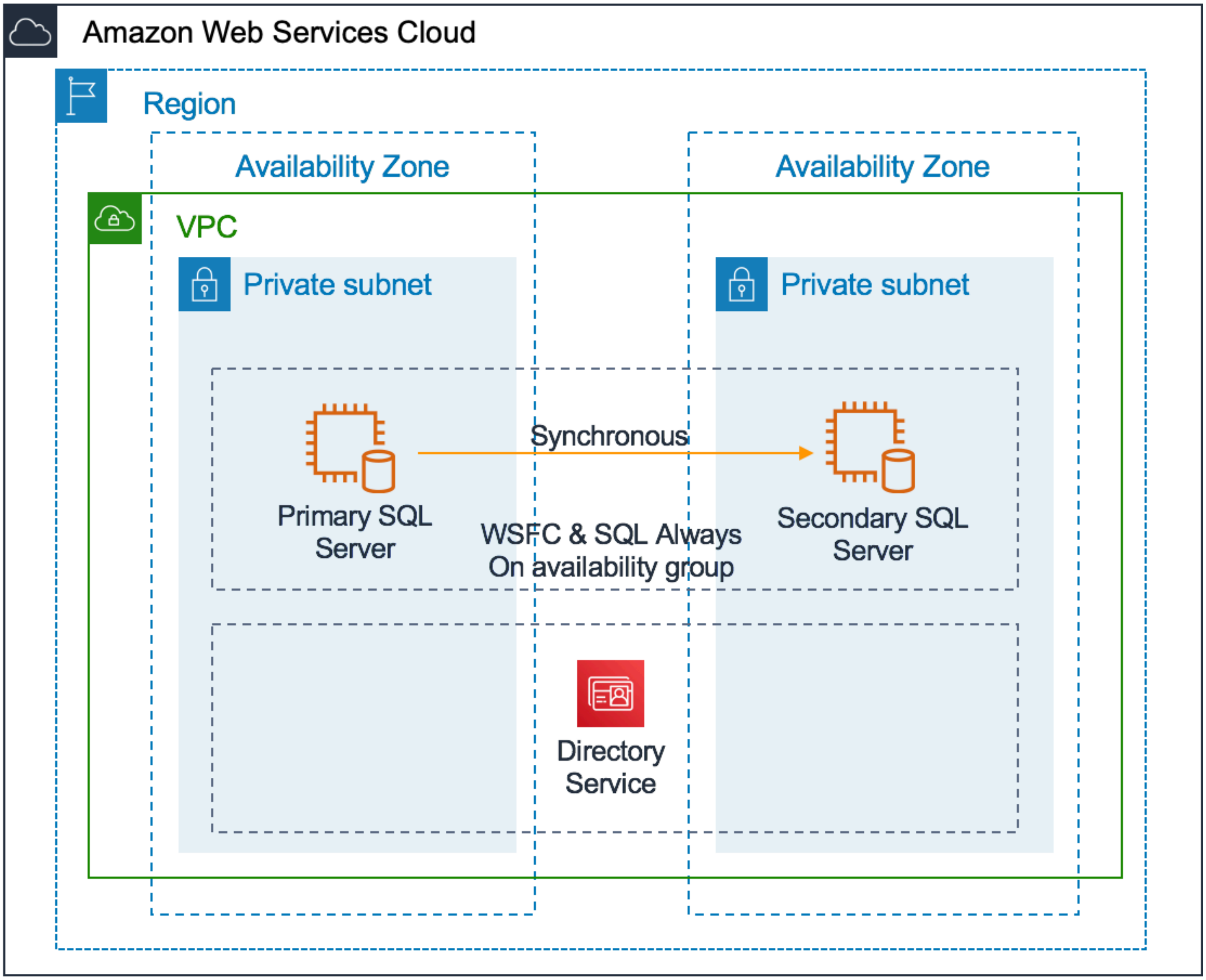
A prerequisite for deploying a SQL Server Always On availability group is Windows Server Failover Clustering (WSFC). SQL Server Always On uses WSFC to increase application availability. WSFC provides infrastructure features that complement the high availability and disaster recovery scenarios supported in the AWS Cloud. Implementing WSFC cluster on AWS is very similar to deploying it on-premises provided you meet two key requirements:

- Deploy the cluster nodes inside an Amazon VPC.
- Deploy the cluster nodes in separate subnets that are in different Availability Zones.

See Overview of Always On Availability Groups (SQL Server) for details.

The following figure provides an overview of architecture for SQL Server Always On availability groups on AWS. This architecture includes following components

- A VPC configured with private subnets across two Availability Zones. This provides the network infrastructure for your SQL Server deployment.
- AWS Directory Service for Microsoft Active Directory deployed in private subnet. Alternatively, you can also manage your own AD DS deployed on Amazon EC2 instance.
- In a private subnet, Windows Servers configured with WSFC for SQL Server Enterprise edition with SQL Server Always On availability groups.



Third-Party Solutions

You can also use third-party tools like SIOS Data Protection Suite, NEC ExpressCluster, or Veritas InfoScale to provide high-availability for SQL Server. These solutions use WSFC and replicate data from primary to secondary with block level replication of the Amazon EBS volume.

Disaster Recovery

Disaster recovery is about preparing for and recovering from a disaster. Any event that has a negative impact on your business continuity or finances could be termed a disaster. To implement a cost effective Disaster recovery strategy for your SAP applications and databases that meets your business objective you need to consider the following requirements.

Separate DR Strategy from HA Design

First you must evaluate whether a separate DR strategy is required in addition to the HA design offered by AWS protection.

On AWS, we recommend that you deploy business critical application in high availability architecture across two Availability Zones in an AWS Region. Each Availability Zone is designed as an independent failure zone. This means that Availability Zones are physically separated within a typical metropolitan region and are located in lower risk flood plains. Availability Zones include a discrete uninterruptable power supply (UPS) and onsite backup generation facilities, and are each fed via different grids from independent utilities to further reduce single points of failure. The level of protection provided by Availability Zone design is sufficient for most customers and is able to meet their business objectives.

DR in AWS Regions

If you determine that you need a separate DR strategy, next you must decide if you need a DR plan in a different AWS Region than your primary AWS Region or in same AWS Region as you primary (for example, using third Availability Zone of your primary AWS Region as DR). Data sovereignty is the primary reason that influences this decision. However, there may be other reasons, such as proximity to users, cost, ease of management, and so on.

DR Architecture

Finally, you must decide on the DR architecture and understand the infrastructure required to implement it. The Recovery Time Objective (RTO) and Recovery Point Objective (RPO) are the primary factors that influence DR architecture. We recommend any of the following three DR architectures:

- **Cold:** This architecture essentially relies on backups. Backups are taken (database – data and log, AMI, Snapshots) on a regular basis and used to rebuild the systems in the target AWS Region to recover for any disaster. Because this architecture completely depends on backups, the RPO depends on how frequently you take backups, and RTO depends on how large the database is to be recovered.
- **Pilot Light:** This option provides better RTO/RPO over cold option because the SQL server database is synchronously or asynchronously getting replicated to a smaller EC2 instance. If you choose this architecture, you mus resize SQL Server EC2 instances, create application server from AMIs before starting production operations. You can use AWS CloudFormation to automate these tasks.

- **Hot DR:** SQL Server database for DR EC2 instances are sized the same as production instances which helps to reduce recovery time over Pilot light because you do not need to resize the instances before starting production operations. For application servers, you can choose to replicate the volumes with CloudEndure or other third-party tools, like SIOS, ATAMotion, and so on.

Depending on your specific RTO/RPO, you can implement cold, pilot light, or hot DR architecture. The following table below provides a comparison between cold and pilot light DR for achievable RTO/RPO.

Table 3 - Cold versus Pilot light DR

DR Architecture	Strategy	RTO/RPO
Cold	SQL Server backup/restore	High/High*
Cold	Amazon AMI	Low/High
Cold	Amazon AMI with frequent DB volumes (Data & Log) snapshots	Low/Low*
Pilot Light	Sync Replication (with-in primary region)	Low/Near-Zero
Pilot Light	Async Replication (in different region)	Low/Few Minutes
Hot	Async Replication (in different region)	Few Minutes/Few Minutes

*The exact time it will take to recover database in DR scenario depends on how much you need to catch up to achieve point in time required for Cold architecture. **High** – couple of hours to a day or more. **Low** –less than an hour to couple of hours.

Deployment

Topics

- Windows EC2 Instance Deployment
- SQL Server Deployment
- SQL Server Deployment for High Availability

Windows EC2 Instance Deployment

Deciding the right storage layout is important to ensure you are able to meet required IO. Amazon EBS general purpose volume (gp2) provides 3 IOPS per GB whereas provisioned IOPS (io1) provide a max of 50 IOPS per GB. See EBS features for details. If you decide to separate SQL data, log, and tempdb to different volumes, consider these aspects.

For gp2, with one volume for all (data, log, and tempdb). Create storage config file as below. Replace placeholder `<size>` as per your requirement.

```
[
    {
        "DeviceName": "xvdb",
        "Ebs": {
            "VolumeSize": <size>,
            "VolumeType": "gp2",
            "DeleteOnTermination": true
        }
    }
]
```

For separate volumes, gp2 (data), io1 (log) and io1 (tempdb) create storage configuration file as below. Replace placeholders `<size>` and `<IOPS Required>` with size of the disk and IOPS you need.

```
[
    {
        "DeviceName": "xvdb",
        "Ebs": {
            "VolumeSize": <size>,
            "VolumeType": "gp2",
            "DeleteOnTermination": true
        }
    },
    {
        "DeviceName": "xvdc",
        "Ebs": {
            "VolumeSize": <size>,
            "VolumeType": “io1",
            "Iops": <IOPS Required>,
            "DeleteOnTermination": true
        }
    },
    {
        "DeviceName": "xvdd",
        "Ebs": {
            "VolumeSize": <size>,
            "VolumeType": “io1",
            "Iops": <IOPS Required>,
            "DeleteOnTermination": true
        }
    }
]
```

SQL Server Deployment

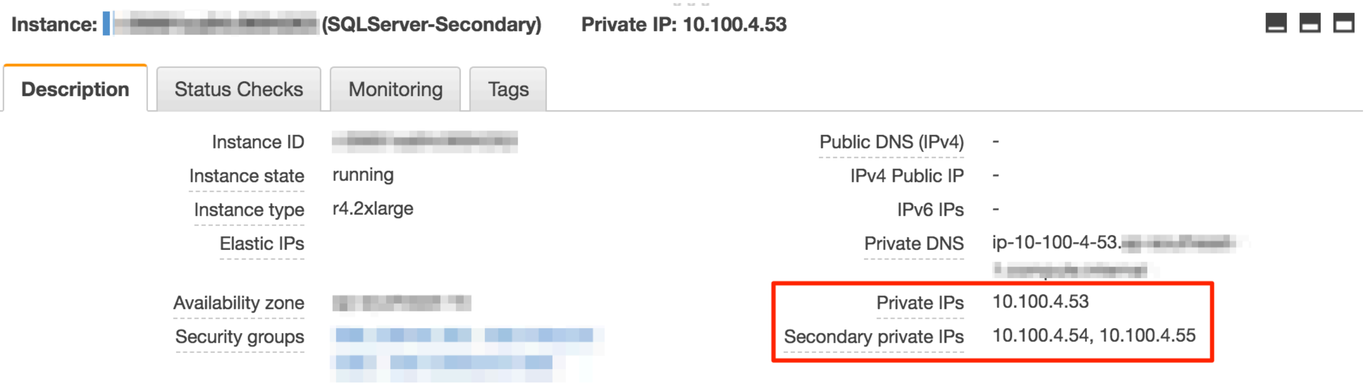
Follow the instructions in the appropriate SAP installation guide for your version of SAP NetWeaver and your combination of operating system and database. See SAP installation guides.

SQL Server Deployment for High Availability

1. Deploy the SAP NetWeaver ASCS instance. For instructions, see the SAP NetWeaver on AWS Deployment and Operations Guide for Windows .
2. Create two EC2 instances for Microsoft SQL server, one in each Availability Zone. See the Windows EC2 instances deployment section for steps.
3. Assign two secondary IP addresses to each instance from the same subnet CIDR in which they are installed:
    a. Use one address for Windows Server Failover Cluster (WSFC).
    b. Use the second address for the Availability Group listener.

       You can assign IP addresses through the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS Tools for Windows PowerShell. For detailed working instructions, see Multiple IP Addresses.

For example, in the screenshot that follows, 10.100.4.53 is the primary private IP address of the EC2 instance. It has been allocated two secondary private addresses: 10.100.4.54 and 10.100.4.55.



Instance: [redacted] (SQLServer-Secondary) Private IP: 10.100.4.53

Instance ID	[redacted]	Public DNS (IPv4)	-
Instance state	running	IPv4 Public IP	-
Instance type	r4.2xlarge	IPv6 IPs	-
Elastic IPs		Private DNS	ip-10-100-4-53.[redacted]
Availability zone	[redacted]	Private IPs	10.100.4.53
Security groups	[redacted]	Secondary private IPs	10.100.4.54, 10.100.4.55

4. Domain join EC2 instances created in Step 1. If you are using AWS Managed Microsoft AD, see AWS Directory Service documentation for detailed steps.
5. Log in to the EC2 instance as admin, open PowerShell, and execute the following command to install the Windows Failover Clustering feature.

```
Install-WindowsFeature -Name Failover-Clustering -restart -IncludeAllSubFeature
```

Note

This command may force your EC2 instance to restart. Make sure you execute the command on both EC2 instances.

6. Log in as domain admin into one of the EC2 instance and execute the following command to create the Windows Server Failover Cluster. Make sure to replace the placeholders before executing the command.

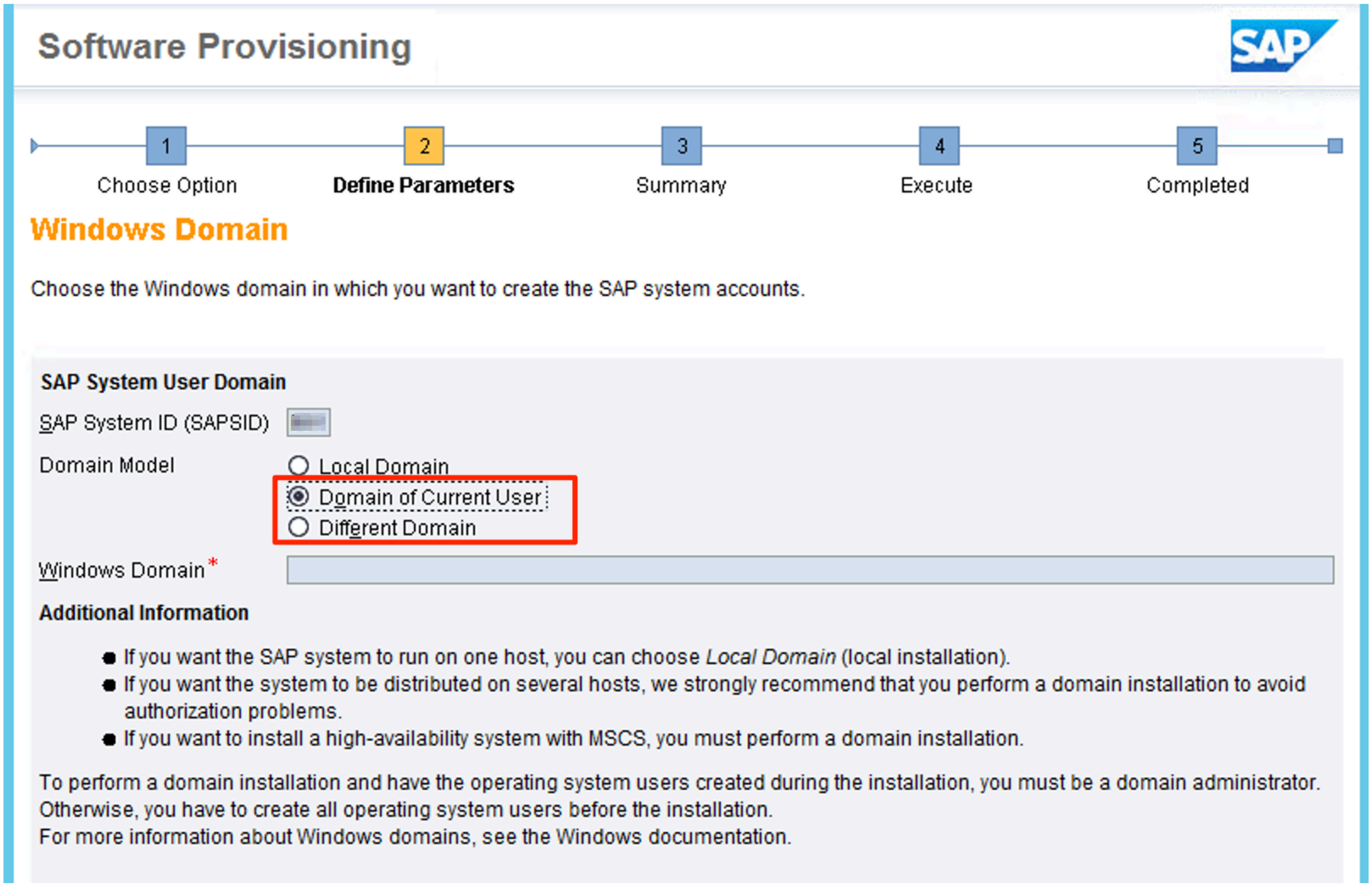
```
New-Cluster -Name <ClusterName> -Node <Node1>,<Node2> -NoStorage
```

For example:

```
New-Cluster -Name SAPSQLCluster -Node primarysql,secondarysql -NoStorage
```

7. Install SQL Server on both EC2 instances. For instructions, see the SAP installation guide.

Install the database instance on the primary node. Follow SAP installation guide to install Database instance on primary node. Make sure you perform domain installations and choose **Domain of Current User** or **Different Domain** as appropriate during parameter selection.



Windows Domain

Choose the Windows domain in which you want to create the SAP system accounts.

SAP System User Domain

SAP System ID (SAPSID) [redacted]

Domain Model
○ Local Domain
● Domain of Current User
○ Different Domain

Windows Domain*

Additional Information

- If you want the SAP system to run on one host, you can choose *Local Domain* (local installation).
- If you want the system to be distributed on several hosts, we strongly recommend that you perform a domain installation to avoid authorization problems.
- If you want to install a high-availability system with MSCS, you must perform a domain installation.

To perform a domain installation and have the operating system users created during the installation, you must be a domain administrator. Otherwise, you have to create all operating system users before the installation.
For more information about Windows domains, see the Windows documentation.

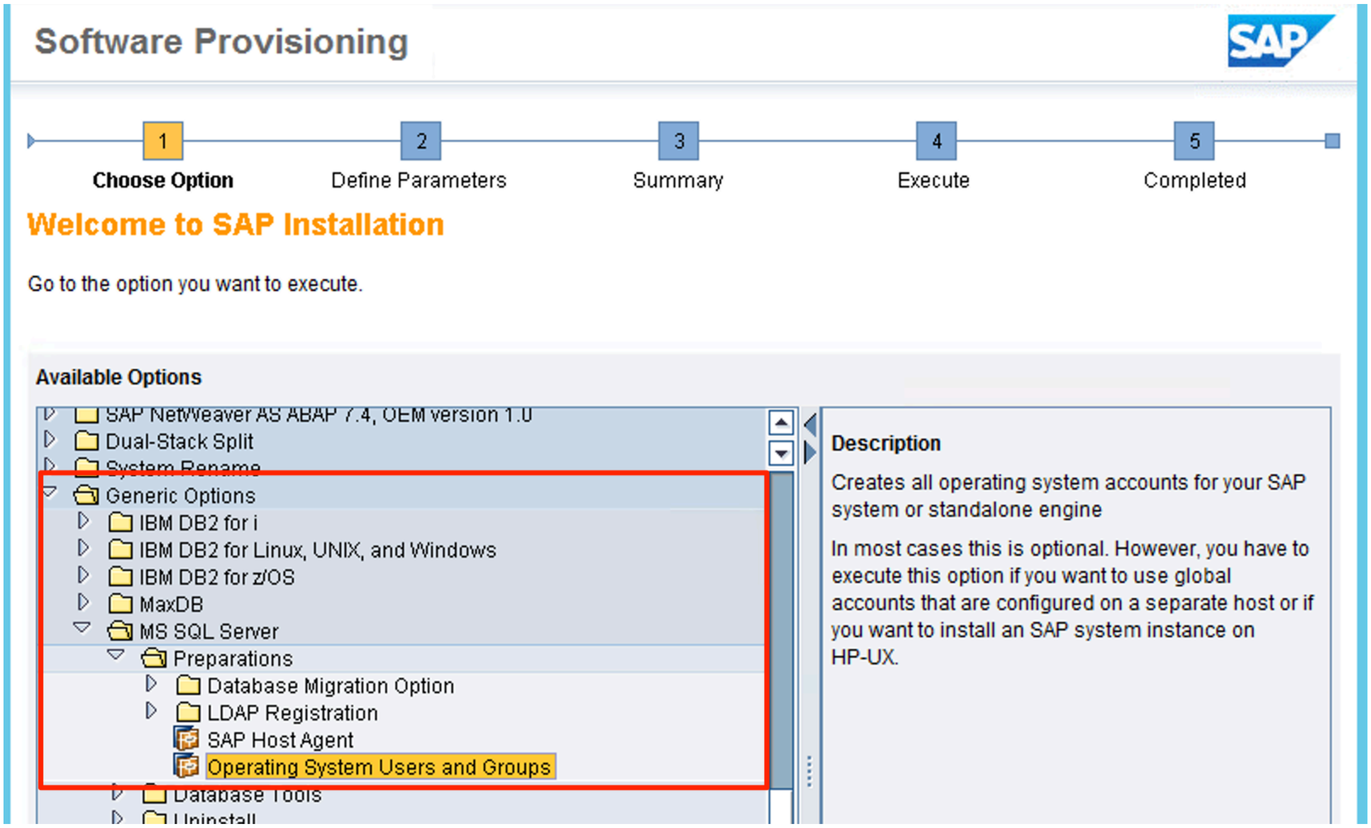
8. Create operating system users on secondary instance.

   a. Start **sapinst**, and in the **Available Options** pane, navigate to **Generic Options > MS SQL Server > Preparations > Operating System Users and Groups**.

   > **Note**
   >
   > The navigation path can vary depending on the version of SWPM you are using.

   b. Create users and groups for this instance, as appropriate.

> **Note**
>
> You do not need to create users on the primary instance because the database instance was installed on the primary node operating system.



Software Provisioning

1 Choose Option — 2 Define Parameters — 3 Summary — 4 Execute — 5 Completed

Welcome to SAP Installation

Go to the option you want to execute.

Available Options

- SAP NetWeaver AS ABAP 7.4, OEM version 1.0
- Dual-Stack Split
- System Rename
- Generic Options
  - IBM DB2 for i
  - IBM DB2 for Linux, UNIX, and Windows
  - IBM DB2 for z/OS
  - MaxDB
  - MS SQL Server
    - Preparations
      - Database Migration Option
      - LDAP Registration
      - SAP Host Agent
      - Operating System Users and Groups
    - Database Tools
    - Uninstall

Description

Creates all operating system accounts for your SAP system or standalone engine

In most cases this is optional. However, you have to execute this option if you want to use global accounts that are configured on a separate host or if you want to install an SAP system instance on HP-UX.

9. Install SAP Host agent on secondary instance with SWPM.
10. Create a SQL Server Always On availability group. See the Microsoft documentation for SQL Always On availability group installation instructions.
11. Adjust the SAP profile files for parameters per the following example. Make sure to replace the `<availabilitygroup listener>` placeholder with appropriate the value for your setup. For details, refer to SAP Note 1772688 - SQL Server Always On and SAP applications.

    ```
    dbs/mss/server = <availabilitygroup listener>;MultiSubnetFailover=yes
    ```

    SAPDBHOST = `<availabilitygroup listener>`
12. Perform failover and failback of SQL Server to validate it is working correctly.

13. Continue with installation of primary application server (PAS) and additional application server (AAS) following the instructions in SAP installation guides.

Operations

This section provides information on AWS services that help you with day-to-day operations of your SQL Server database for SAP applications.

Topics

Monitoring

AWS provides multiple services to monitor and manage your infrastructure and applications on AWS. You can use services like Amazon CloudWatch and AWS CloudTrail to monitor your underlying infrastructure and APIs, respectively.

CloudWatch provides ready-to-use key performance indicators (KPIs) that you can use to monitor both CPU and disk utilization.

You can also create custom metrics for monitoring SQL server.

With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. AWS CloudTrail is enabled on all AWS accounts and records your account activity upon account creation. You can view and download the last 90 days of your account activity for create, modify, and delete operations of supported services without the need to manually set up CloudTrail.

Backup and Recovery

You need to regularly back up your operating system and database to recover them in case of any failure. AWS provides various services and tools that you can use to back up your SQL Server database of SAP applications.

Amazon Machine Images (AMIs)

You can use the AWS Management Console or the AWS CLI to create a new AMI of your existing SAP system. This AMI can be used to recover your existing SAP system or to create a clone.

The AWS CLI `create image` command creates a new AMI based on an existing Amazon EC2 instance. The new AMI contains a complete copy of the operating system and its configuration, software configurations, and optionally all Amazon EBS volumes that are attached to the instance. For details on how to create an AMI of an existing Amazon EC2 instance, see Creating an Amazon EBS Backed Windows AMI. AMI creation and lifecycle can be centrally managed in AWS Backup AWS Backup.

Amazon EBS Snapshots

You can back up your Amazon EBS volumes to Amazon Simple Storage Service by taking point-in-time snapshots. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved.

Snapshots are suited to back up SAP file systems like `/usr/sap/` , `/sapmnt/`. If you decide to take snapshots of your EBS volumes containing data and log files, make sure to use Volume Shadow Copy Service or shut down your database before a snapshot is triggered for consistency. You can use AWS Backup to create backups using VSS functionalities.

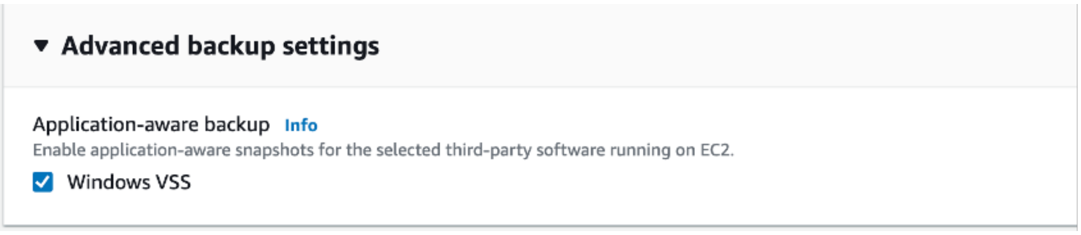
The following command creates a snapshot of volume (with example `volume id vol-1234567890abcdef0`). You can use this command in AWS CLI to create your own volume snapshot.

```
aws ec2 create-snapshot --volume-id <vol-1234567890abcdef0> --description "This is my
 volume snapshot."
```

Database Backups

For SQL Server database backup, you can use one of the following methods:

- **SQL native tools to take backup on disk:** Backup requires high throughput compared to IOPS. We recommend using Throughput Optimized HDD (st1) which provides maximum throughput of 500 MB/s per volume. Once the backup completes on disk, you can use scripts to move it to an Amazon S3 bucket.
- **AWS Backup** for application-consistent backups via Microsoft's Volume Shadow Copy Services (VSS). Ensure that the flag in the advanced backup settings is enabled:



▼ Advanced backup settings

Application-aware backup Info

Enable application-aware snapshots for the selected third-party software running on EC2.

☑ Windows VSS

- **Third-party backint tools:** Partners like Commvault, Veritas, and so on use SAP backint interface and store backups directly in Amazon S3 buckets.

Storage

The following list includes Amazon storage services included in this guide.

Amazon EBS

Amazon EBS provides persistent storage for SAP application and database. You can increase EBS volume size or change the type of volume (for example, gp2 to io1) without requiring downtime. For more information, see Modifying Amazon EBS volume.

Amazon FSx for Windows File Server

Amazon FSx does not require you to explicitly provision storage at all – you simply pay for what you use.

Amazon FSx requires regular maintenance for patching, but you can define the maintenance windows as per your business requirements. For details, see FSx Maintenance Windows.

Amazon S3

Amazon S3 does not require you to explicitly provision storage at all – you simply pay for what you use.

Operating System Maintenance

In general, operating system maintenance across large estates of EC2 instances can be managed by:

- Tools specific to each operating system, such as Microsoft System Center 2019
- Third-party products, such as those available on AWS Marketplace
- AWS Systems Manager

AWS Systems Manager can help with the following key operating system maintenance tasks.

Patching

You can follow SAP recommended patching processes to update your landscape on AWS. For operating system patching, use AWS Systems Manager Patch Manager to roll out OS patches as per your corporate policies. Patch manager includes features like:

- Scheduling based on tags
- Auto-approving patches with lists of approved and rejected patches
- Defining patch baselines

AWS Systems Manager Patch Manager integrates with AWS Identity and Access Management (IAM), AWS CloudTrail, and Amazon CloudWatch Events to provide a secure patching experience that includes event notifications and the ability to audit usage. For details about the process, see How Patch Manager Operations Work. If AWS Systems Manager Patch Manager does not fulfil your requirements, there are third-party products available on the AWS Marketplace.

Maintenance Window

AWS Systems Manager Maintenance Windows let you define a schedule for when to perform potentially disruptive actions on your instances, such as patching an operating system, updating drivers, or installing software or patches.

Automation using Documents

AWS Systems Manager Automation simplifies common maintenance and deployment tasks of Amazon EC2 instances and other AWS resources. Automation enables you to do the following:

- Build Automation workflows to configure and manage instances and AWS resources.
- Create custom workflows or use pre-defined workflows maintained by AWS.
- Receive notifications about Automation tasks and workflows by using Amazon CloudWatch Events.
- Monitor Automation progress and execution details by using the Amazon EC2 or the AWS Systems Manager console.

There are many AWS provided documents specific to Windows already available.

Business Continuity

AWS recommends that you periodically schedule business continuity process validations by executing disaster recovery (DR) tests. This planned activity will help to flush out any potential unknowns and help the organization to deal with any real disaster in a streamlined manner. Depending on your disaster recovery architecture, business continuity may include:

- Backup/recovery of database from AmazonS3
- Creation of systems from AMI and point-in-time recovery via snapshots
- Changing the EC2 instance size of pilot light system
- Validation of integration (AD/DNS, email, third party, and so on.)

Support

SAP requires customers to have a minimum of an AWS Business Support plan with AWS. This ensures that any critical issues raised with SAP are also handled by AWS on priority. AWS Business Support provides less than one hour response time for production down scenarios. For a response time of less than 15 minute for business critical systems along with other benefits, you can choose AWS Enterprise Support.

For any SAP application issues, AWS suggests that you raise an incident with SAP via the SAP Support portal. After the first level of investigation, SAP can redirect the incident to AWS Support if the issue is infrastructure-related. However, if you choose to raise support issues for SAP applications with AWS Support, we cannot redirect the tickets to SAP. For any infrastructure-related issues, you can raise the issue directly with AWS Support.

Cost Optimization

Resources (CPU, Memory, additional application servers, system copies for different tests/ validations, and so on) require SAP landscape changes over time. AWS recommends that you monitor system utilization and the need for existing systems on a regular basis and take actions to reduce cost. In case of a database like SQL Server, the only opportunity to right-size the database server is by scaling up/down or shutting it down, if not required. Here are few suggestions that you can consider for cost optimization:

- Consider Reserved instances over On-Demand instances if the requirement is to run your instances 24x7 365 days per year. Reserved instances provide up to a 75% discount over On-Demand instances. See EC2 pricing for details.
- Consider running occasionally required systems like training, sandbox, and so on, on-demand for the duration required.
- Monitor CPU and memory utilization over time for other non-production systems like Dev/QA and right-size them when possible.

FAQ

Q. Can I use Amazon RDS for SQL Server as a database to deploy SAP NetWeaver based applications?

A. No, Amazon RDS for SQL Server is not certified by SAP for SAP NetWeaver based applications. However, it is certified to be used as database for SAP Business Objects BI (BObj BI)

Q. Can I purchase and use a Microsoft SQL Server license from AWS, such as Microsoft SQL Server 2019 Enterprise on Windows Server 2022, Amazon Machine Image (AMI), to host my SAP NetWeaver based workloads, and other SAP workloads?

A. Yes, AWS provides a variety of options for Microsoft SQL Server license-included AMIs, as a pre-installed package with different combinations of Microsoft Windows Server and Microsoft SQL Server versions and editions available. For more information, see Licensing options and Find a SQL Server license-included AMI.

There are some differences in how SAP manages technical support, when the support ticket is raised with SAP support, and if the issue raised is found to be with Microsoft SQL Server, when those licenses are from AWS. In that situation, you need to raise a separate ticket with Support for SQL Server technical support, following the terms of your Support plan.

Document Revisions

Date	Change
December 2020	Minor updates to text in Backup & Recovery section
July 2019	Initial publication

SAP NetWeaver on AWS: high availability configuration for SUSE Linux Enterprise Server (SLES) for SAP applications

This topic applies to SUSE Linux Enterprise Server (SLES) operating system for SAP NetWeaver applications on AWS cloud. It covers the instructions for configuration of a pacemaker cluster for the ABAP SAP Central Service (ASCS) and the Enqueue Replication Server (ERS) when deployed on Amazon EC2 instances in two different Availability Zones within an AWS Region.

This topic covers instructions for the following configuration options.

Topics

Planning

This section covers the following topics.

Topics

Prerequisites

You must meet the following prerequisites before commencing setup.

Topics

Deployed cluster infrastructure

Ensure that your AWS networking requirements and Amazon EC2 instances where SAP workloads are installed, are correctly configured for SAP. For more information, see SAP NetWeaver Environment Setup for Linux on AWS.

See the following ASCS cluster specific requirements.

- Two cluster nodes created in private subnets in separate Availability Zones within the same Amazon VPC and AWS Region
- Access to the route table(s) that are associated with the chosen subnets

  For more information, see AWS – Overlay IP.
- Targeted Amazon EC2 instances must have connectivity to the Amazon EC2 endpoint via internet or a Amazon VPC endpoint.

Supported operating system

Protecting ASCS with a pacemaker cluster requires packages from SUSE, including targeted cluster resource agents for SAP and AWS that may not be available in standard repositories.

For deploying SAP applications on SUSE, SAP and SUSE recommend using SUSE Linux Enterprise Server for SAP applications (SLES for SAP). SLES for SAP provides additional benefits, including Extended Service Pack Overlap Support (ESPOS), configuration and tuning packages for SAP applications, and High Availability Extensions (HAE). For more details, see SUSE website at SUSE Linux Enterprise Server for SAP Applications.

SLES for SAP is available at AWS Marketplace with an hourly or annual subscription. You can also use the bring your own subscription (BYOS) model.

Required access for setup

The following access is required for setting up the cluster.

- An IAM user with the following privileges.
  - modify Amazon VPC route tables
  - modify Amazon EC2 instance properties
  - create IAM policies and roles
  - create Amazon EFS file systems
- Root access to the operating system of both cluster nodes
- SAP administrative user access – `<sid>adm`

  In case of a new install, this user is created by the install process.

Reliability

The SAP Lens of the Well-Architected framework, in particular the Reliability pillar, can be used to understand the reliability requirements for your SAP workload.

ASCS is a single point of failure in a highly available SAP architecture. The impact of an outage of this component must be evaluated against factors, such as, recovery point objective (RPO), recovery time objective (RTO), cost and operation complexity. For more information, see Reliability in SAP Lens - AWS Well-Architected Framework.

SAP and SUSE references

In addition to this guide, see the following references for more details.

- SUSE documentation – SAP S/4 HANA - Enqueue Replication 2 High Availability Cluster With Simple Mount
- SUSE documentation – SAP S/4 HANA - Enqueue Replication 2 High Availability Cluster
- SAP Note: 1656099 - SAP Applications on AWS: Supported DB/OS and Amazon EC2 products
- SAP Note: 1984787 - SUSE Linux Enterprise Server 12: Installation Notes
- SAP Note: 2578899 - SUSE Linux Enterprise Server 15: Installation Notes
- SAP Note: 1275776 - Linux: Preparing SLES for SAP environments

You must have SAP portal access for reading all SAP Notes.

Concepts

This section covers AWS, SAP, and SUSE concepts.

Topics

SAP – ABAP SAP Central Services (ASCS)

The ABAP SAP Central Services (ASCS) is an SAP instance consisting of the following two services. It is considered a single point of failure (SPOF) in a resilient SAP system.

- **Message server** – Responsible for application load distribution (GUI and RFC), communication between application servers, and centralised configuration information for web dispatchers and application servers.
- **Enqueue server (standalone)** – Maintains a lock table in main memory (shared memory). Unlike a database lock, an enqueue lock can exist across multiple logical units of work (LUW), and is set by a SAP Dialog work process. The lock mechanism prevents two transactions from changing the same data in the database simultaneously.

> **Note**
>
> With ABAP Release 7.53 (ABAP Platform 1809), the new Standalone Enqueue Server 2 (ENSA2) is installed by default. It replaces the previous version (ENSA1) but can be configured for the previous versions. See SAP Note 2630416 - Support for Standalone Enqueue Server 2 (SAP portal access required) for more information.
> This document includes modifications to align with the correct ENSA version.

SAP – Enqueue Replication Server (ERS)

The Enqueue Replication Server (ERS) is an SAP instance containing a replica of the lock table (replication table).

In a resilient setup, if the standalone enqueue server (EN/ENQ) fails, it can be restarted either by restart parameters or by high availability software, such as Pacemaker. The enqueue server retrieves the replication table remotely or by failing over to the host where the ERS is running.

AWS – Availability Zones

Availability Zone is one or more discreet data centers with redundant power, networking, and connectivity in an AWS Region. For more information, see Regions and Availability Zones.

For mission critical deployments of SAP on AWS where the goal is to minimise the recovery time objective (RTO), we suggest distributing single points of failure across Availability Zones. Compared with single instance or single Availability Zone deployments, this increases resilience and isolation against a broad range of failure scenarios and issues, including natural disasters.

Each Availability Zone is physically separated by a meaningful distance (many kilometers) from another Availability Zone. All Availability Zones in an AWS Region re interconnected with high-bandwidth, low-latency network, over fully redundant, dedicated metro fiber. This enables synchronous replication. All traffic between Availability Zones is encrypted.

AWS – Overlay IP

Overlay IP enables a connection to the application, regardless of which Availability Zone (and subnet) contains the active primary node.

When deploying an Amazon EC2 instance in AWS, IP addresses are allocated from the CIDR range of the allocated subnet. The subnet cannot span across multiple Availability Zones, and therefore the subnet IP addresses may be unavailable after faults, including network connectivity or hardware issues which require a failover to the replication target in a different Availability Zone.

To address this, we suggest that you configure an overlay IP, and use this in the connection parameters for the application. This IP address is a non-overlapping RFC1918 private IP address from outside of VPC CIDR block and is configured as an entry in the route table or tables. The route directs the connection to the active node and is updated during a failover by the cluster software.

You can select any one of the following RFC1918 private IP addresses for your overlay IP address.

- 10.0.0.0 – 10.255.255.255 (10/8 prefix)
- 172.16.0.0 – 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 – 192.168.255.255 (192.168/16 prefix)

If, for example, you use the 10/8 prefix in your SAP VPC, selecting a 172 or a 192 IP address may help to differentiate the overlay IP. Consider the use of an IP Address Management (IPAM) tool such as Amazon VPC IP Address Manager to plan, track, and monitor IP addresses for your AWS workloads. For more information, see What is IPAM?

The overlay IP agent in the cluster can also be configured to update multiple route tables which contain the Overlay IP entry if your subnet association or connectivity requires it.

Access to overlay IP

The overlay IP is outside of the range of the VPC, and therefore cannot be reached from locations that are not associated with the route table, including on-premises and other VPCs.

Use AWS Transit Gateway as a central hub to facilitate the network connection to an overlay IP address from multiple locations, including Amazon VPCs, other AWS Regions, and on-premises using AWS Direct Connect or AWS Client VPN.

If you do not have AWS Transit Gateway set up as a network transit hub or if it is not available in your preferred AWS Region, you can use a Network Load Balancer to enable network access to an overlay IP.

For more information, see SAP on AWS High Availability with Overlay IP Address Routing.

AWS – Shared VPC

An enterprise landing zone setup or security requirements may require the use of a separate cluster account to restrict the route table access required for the Overlay IP to an isolated account. For more information, see Share your VPC with other accounts.

Evaluate the operational impact against your security posture before setting up shared VPC. To set up, see Shared VPC – optional.

Pacemaker - STONITH fencing agent

In a two-node cluster setup for a primary resource and its replication pair, it is important that there is only one node in the primary role with the ability to modify your data. In the event of a failure scenario where a node is unresponsive or incommunicable, ensuring data consistency requires that the faulty node is isolated by powering it down before the cluster commences other actions, such as promoting a new primary. This arbitration is the role of the fencing agent.

Since a two-node cluster introduces the possibility of a fence race in which a dual shoot out can occur with communication failures resulting in both nodes simultaneously claiming, "I can’t see you, so I am going to power you off". The fencing agent is designed to minimise this risk by providing an external witness.

SLES supports several fencing agents, including the one recommended for use with Amazon EC2 Instances (external/ec2). This resource uses API commands to check its own instance status - "Is my instance state anything other than running?" before proceeding to power off its pair. If it is already in a stopping or stopped state it will admit defeat and leave the surviving node untouched.

Pacemaker - simple-mount and classic architecture

This guide covers two architectures for SAP cluster solutions on SLES for SAP – simple-mount and classic (previous standard). Simple-mount was certified as the SLES for SAP Applications cluster solution in late 2021. It is now the recommended architecture for both ENSA1 and ENSA2 deployments running on SLES for SAP 15 and above. For more details, see SUSE blog Simple Mount Structure for SAP Application Platform.

If you are configuring a new SAP installation, we recommend the simple-mount architecture. If you already have the classic architecture, and wish to migrate to the simple-mount architecture, see Switching architecture to simple-mount.

The following are the differences between the classic and simple-mount architectures.

- Removing file system resources from cluster – a file system is required but it is not mounted and unmounted by the cluster. The executable directory for the ASCS and ERS can be permanently mounted on both nodes.
- Addition of `SAPStartSrv` – `SAPStartSrv` controls the matching `SAPStartSrv` framework process.
- Sapping and sappong services – these services manage the start of `SAPStartSrv` services with sapinit.

See the Architecture diagrams for more details.

Automation and AWS Launch Wizard for SAP

You can set up a cluster manually using the instructions provided here. You can also automate parts of this process to ensure consistency and repeatability.

Use AWS Launch Wizard for SAP for automated deployments of SAP NetWeaver, SAP S/4 HANA, SAP B/4HANA, and Solution Manager. Launch Wizard uses AWS CloudFormation scripts to quickly provision the resources needed to deploy SAP NetWeaver and S/4 HANA. The automation performs SAP enqueue replication and pacemaker setup so that only validation and testing are required. For more information, see AWS Launch Wizard for SAP.

To ensure that the behavior and operation of your cluster is well understood regardless of how your system is set up, we recommend a thorough test cycle. See Testing for more details.

Switching architecture to simple-mount

Follow along these steps if you want to switch an existing cluster with classic architecture to use the recommended configuration of simple-mount architecture.

These steps must be performed in an outage window, allowing stop/start of services and basic testing.

1. Put the cluster in maintenance mode. See Maintenance mode.
2. Stop SAP services, including application servers connected to the cluster as well as ASCS and ERS.
3. Install any missing operating system packages. See Package.

   It might be necessary to install `sapstartsrv-resource-agents`. However, all operating system prerequisites must be checked and updated to ensure that versions are compatible. See Operating system prerequisites.
4. Add entries for ASCS and ERS mount point on both nodes (if not already added). See Update /etc/fstab.
5. Enable `sapping/sappong` services. See Enable sapping/sappongsystemd services (simple-mount only).
6. Disable `systemd` services. See Align and disable SAP auto start services for systemd.
7. Backup the configuration with the following command.

   ```
   crm config show >> /tmp/classic_ha_setup.txt
   ```

See Prepare for resource creation.

8. *Optional* – delete the configuration. You can edit in place but we recommend starting with a blank configuration. This ensures that latest timeout and priority parameters are in place.

```
crm config erase
crm config show
```

9. Configure cluster resources again. See Create cluster resources. Ignore the sections pertaining to the classic architecture.

10. Check the cluster and perform some tests. See Testing.

11. Resume standard operations by starting any additional services, including application servers.

Architecture diagrams

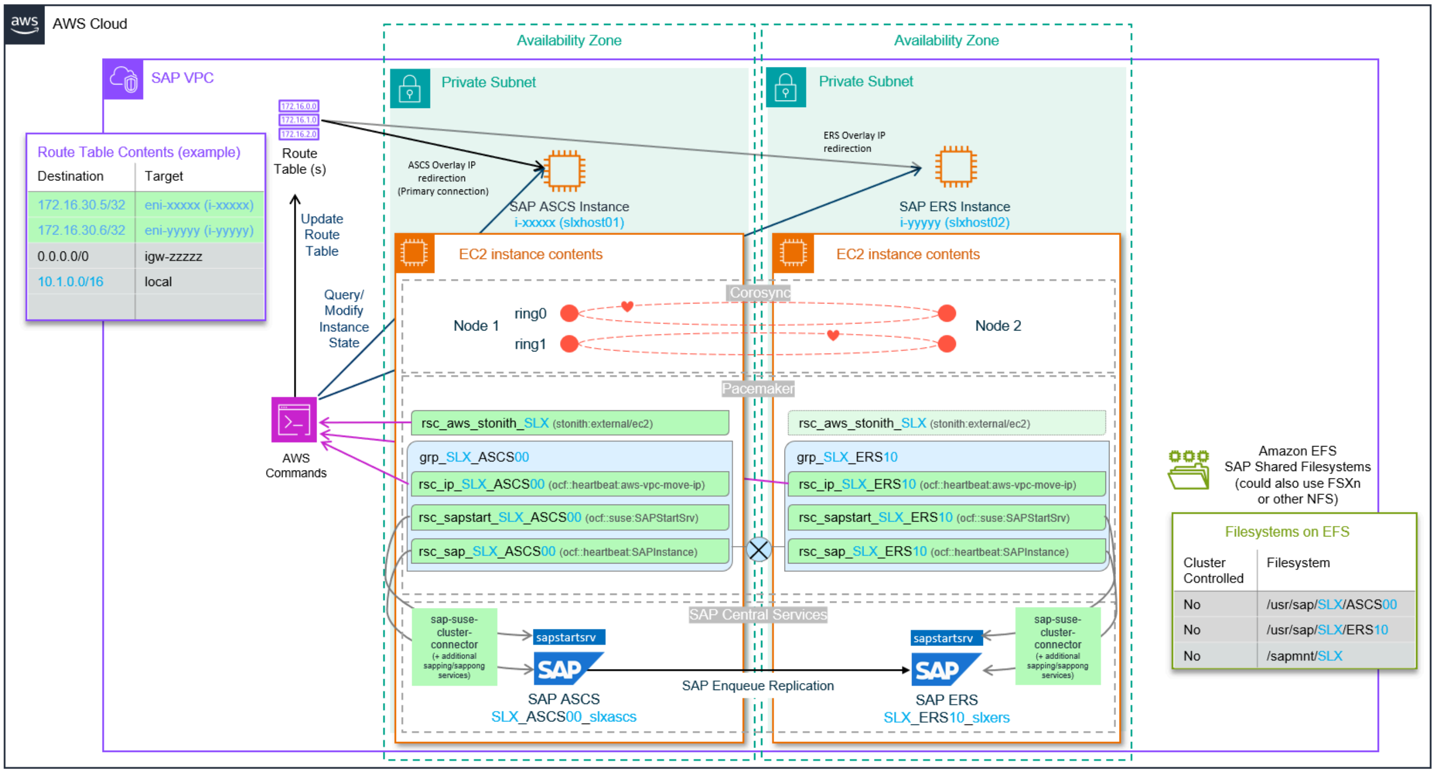
This guide covers two architectures for SAP cluster solutions on SLES for SAP – simple-mount and classic (previous standard). See the following images to learn more.

Topics

- Pacemaker - simple-mount architecture
- Pacemaker - classic architecture

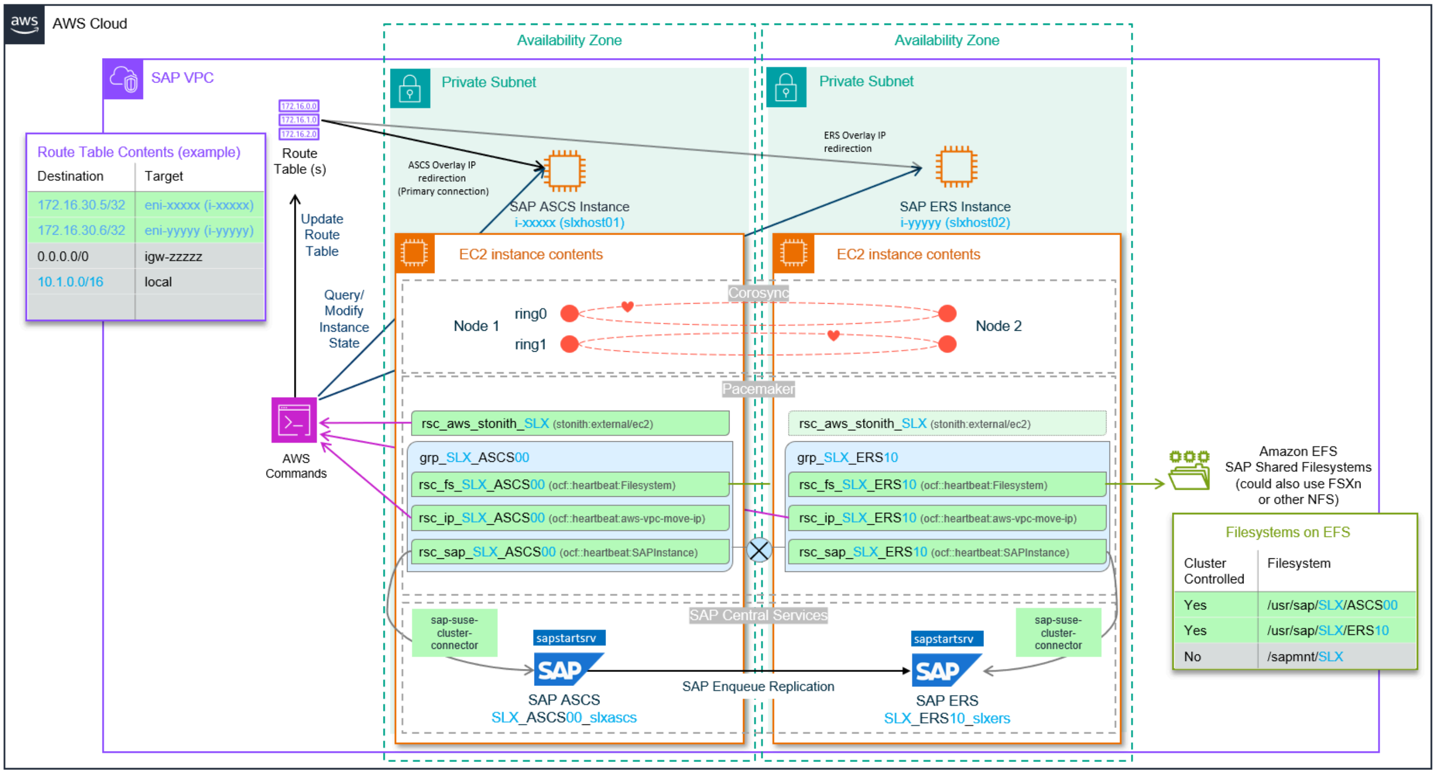
Pacemaker - simple-mount architecture

See the following image for more details.


AWS Cloud
SAP VPC
Availability Zone
Availability Zone
Private Subnet
Private Subnet
172.16.0.0
172.16.1.0
172.16.2.0
Route Table (s)
Route Table Contents (example)
Destination
Target
172.16.30.5/32
eni-xxxxx (i-xxxxx)
172.16.30.6/32
eni-yyyyy (i-yyyyy)
0.0.0.0/0
igw-zzzzz
10.1.0.0/16
local
Update Route Table
Query/ Modify Instance State
AWS Commands
ASCS Overlay IP redirection (Primary connection)
ERS Overlay IP redirection
SAP ASCS Instance
i-xxxxx (slxhost01)
SAP ERS Instance
i-yyyyy (slxhost02)
EC2 instance contents
EC2 instance contents
Corosync
Node 1
ring0
ring1
Node 2
Pacemaker
rsc_aws_stonith_SLX (stonith:external/ec2)
rsc_aws_stonith_SLX (stonith:external/ec2)
grp_SLX_ASCS00
rsc_ip_SLX_ASCS00 (ocf::heartbeat:aws-vpc-move-ip)
rsc_sapstart_SLX_ASCS00 (ocf::suse:SAPStartSrv)
rsc_sap_SLX_ASCS00 (ocf::heartbeat:SAPInstance)
grp_SLX_ERS10
rsc_ip_SLX_ERS10 (ocf::heartbeat:aws-vpc-move-ip)
rsc_sapstart_SLX_ERS10 (ocf::suse:SAPStartSrv)
rsc_sap_SLX_ERS10 (ocf::heartbeat:SAPInstance)
SAP Central Services
sap-suse-cluster-connector (+ additional sapping/sappong services)
sapstartsrv
SAP ASCS
SLX_ASCS00_slxascs
SAP Enqueue Replication
sapstartsrv
SAP ERS
SLX_ERS10_slxers
sap-suse-cluster-connector (+ additional sapping/sappong services)
Amazon EFS
SAP Shared Filesystems (could also use FSXn or other NFS)
Filesystems on EFS
Cluster Controlled
Filesystem
No
/usr/sap/SLX/ASCS00
No
/usr/sap/SLX/ERS10
No
/sapmnt/SLX

Pacemaker - classic architecture

See the following image for more details.



Deployment

This section covers the following topics.

Topics

Settings and prerequisites

The cluster setup uses parameters, including `SID` and `System Number` that are unique to your setup. It is useful to predetermine the values with the following examples and guidance.

Topics

Define reference parameters for setup

The cluster setup relies on the following parameters.

Topics

Global AWS parameters

Name	Parameter	Example
AWS account ID	`<account_id>`	`123456789100`
AWS Region	`<region_id>`	`us-east-1`

- AWS account – For more details, see Your AWS account ID and its alias.
- AWS Region – For more details, see Describe your Regions.

Amazon EC2 instance parameters

Name	Parameter	Primary example	Secondary example
Amazon EC2 instance ID	`<instance_id>`	`i-xxxxinstidforhost1`	`i- xxxxinstidforhost2`
Hostname	`<hostname>`	`slxhost01`	`slxhost02`
Host IP	`<host_ip>`	`10.1.10.1`	`10.1.20.1`
Host additional IP	`<host_additional_ip>`	`10.1.10.2`	`10.1.20.2`
Configured subnet	`<subnet_id>`	`subnet-xxxxxxxxxxsubnet1`	`subnet-xxxxxxxxxxsubnet2`

- Hostname – Hostnames must comply with SAP requirements outlined in SAP Note 611361 - Hostnames of SAP ABAP Platform servers (requires SAP portal access).

  Run the following command on your instances to retrieve the hostname.

  ```
  hostname
  ```

- Amazon EC2 instance ID – run the following command (IMDSv2 compatible) on your instances to retrieve instance metadata.

  ```
  /usr/bin/curl --noproxy '*' -w "\n" -s -H "X-aws-ec2-metadata-token: $(curl --noproxy
   '*' -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
  token-ttl-seconds: 21600")" http://169.254.169.254/latest/meta-data/instance-id
  ```

  For more details, see Retrieve instance metadata and Instance identity documents.

SAP and Pacemaker resource parameters

Name	Parameter	Example
SID	`<SID>` or `<sid>`	`SLX`
ASCS Alias	`<ascs_virt_hostname>`	`slxascs`
ASCS System Number	`<ascs_sys_nr>`	`00`
ASCS Overlay IP	`<ascs_oip>`	`172.16.30.5`
ASCS NFS Mount Point	`<ascs_nfs_mount_point>`	`/SLX_ASCS00`
ERS Alias	`<ers_virt_hostname>`	`slxers`
ERS System Number	`<ers_sys_nr>`	`10`
ERS Overlay IP	`<ers_oip>`	`172.16.30.6`
ERS NFS Mount Point	`<ers_nfs_mount_point>`	`/SLX_ERS10`
ENSA Type	`<ensa_type>`	`ENSA2`
VPC Route Tables	`<rtb_id>`	`rtb-xxxxxroutetable1`
Sapmnt NFS ID or CNAME	`<sapmnt_nfs_id>`	`fs-xxxxxxxxxxxxxefs1`

- SAP details – SAP parameters, including SID and instance number must follow the guidance and limitations of SAP and Software Provisioning Manager. Refer to SAP Note 1979280 - Reserved SAP System Identifiers (SAPSID) with Software Provisioning Manager for more details.

  Post-installation, use the following command to find the details of the instances running on a host.

  ```
  sudo /usr/sap/hostctrl/exe/saphostctrl -function ListInstances
  ```
- Overlay IP – This value is defined by you. For more information, see Overlay IP.
- NFS mount points – This value is defined by you. Consider which systems are going to share an NFS file system (Amazon EFS or Amazon FSx), and ensure that your naming standards allow it.

SLES cluster parameters

Name	Parameter	Example
Cluster user	`cluster_user`	`hacluster`
Cluster password	`cluster_password`	
Cluster tag	`cluster_tag`	`pacemaker`
AWS CLI cluster profile	`aws_cli_cluster_profile`	`cluster`
Cluster connector	`cluster_connector`	`sap-suse-cluster-connector`

Amazon EC2 instance settings

Amazon EC2 instance settings can be applied using Infrastructure as Code or manually using AWS Command Line Interface or AWS Management Console. We recommend Infrastructure as Code automation to reduce manual steps, and ensure consistency.

Topics

- Create IAM roles and policies
- AWS Overlay IP policy

- Assign IAM role
- Modify security groups for cluster communication
- Disable source/destination check
- Review automatic recovery and stop protection
- Create Amazon EC2 resource tags used by Amazon EC2 STONITH agent

Create IAM roles and policies

In addition to the permissions required for standard SAP operations, two IAM policies are required for the cluster to control AWS resources on ASCS. These policies must be assigned to your Amazon EC2 instance using an IAM role. This enables Amazon EC2 instance, and therefore the cluster to call AWS services.

Create these policies with least-privilege permissions, granting access to only the specific resources that are required within the cluster. For multiple clusters, you need to create multiple policies.

For more information, see IAM roles for Amazon EC2.

STONITH policy

The SLES STONITH resource agent (`external/ec2`) requires permission to start and stop both the nodes of the cluster. Create a policy as shown in the following example. Attach this policy to the IAM role assigned to both Amazon EC2 instances in the cluster.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeTags"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": [
        "arn:aws:ec2:<region>:<account_id>:instance/<instance_id_1>",
        "arn:aws:ec2:<region>:<account_id>:instance/<instance_id_2>"
      ]
    }
  ]
}
```

AWS Overlay IP policy

The SLES Overlay IP resource agent (`aws-vpc-move-ip`) requires permission to modify a routing entry in route tables. Create a policy as shown in the following example. Attach this policy to the IAM role assigned to both Amazon EC2 instances in the cluster.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:ReplaceRoute",
            "Resource": [
                 "arn:aws:ec2:<region>:<account_id>:route-table/<rtb_id_1>",
                 "arn:aws:ec2:<region>:<account_id>:route-table/<rtb_id_2>"
                        ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeRouteTables",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> If you are using a Shared VPC, see Shared VPC – optional.

Assign IAM role

The two cluster resource IAM policies must be assigned to an IAM role associated with your Amazon EC2 instance. If an IAM role is not associated to your instance, create a new IAM role for cluster operations. To assign the role, go to https://console.aws.amazon.com/ec2/, select your instance(s), and then choose **Actions** > **Security** > **Modify IAM role**.

Modify security groups for cluster communication

A security group controls the traffic that is allowed to reach and leave the resources that it is associated with. For more information, see Control traffic to your AWS resources using security groups.

In addition to the standard ports required to access SAP and administrative functions, the following rules must be applied to the security groups assigned to both Amazon EC2 instances in the cluster.

Inbound

Source	Protocol	Port range	Description
The security group ID (its own resource ID)	**UDP**	5405	Allows UDP traffic between cluster resources for corosync communica tion
Bastion host security group or CIDR range for administration	**TCP**	7630	*optional* Used for SLES Hawk2 Interface for monitoring and administration using a Web Interface For more informati on, see Configuri ng and Managing Cluster Resources with Hawk2 in the SUSE documentation.

Note

Note the use of the UDP protocol.

If you are running a local firewall, such as `iptables`, ensure that communication on the preceding ports is allowed between two Amazon EC2 instances.

Disable source/destination check

Amazon EC2 instances perform source/destination checks by default, requiring that an instance is either the source or the destination of any traffic it sends or receives.

In the pacemaker cluster, source/destination check must be disabled on both instances receiving traffic from the Overlay IP. You can disable check using AWS CLI or AWS Management Console.

Example

AWS CLI

Use the modify-instance-attribute command to disable source/destination check.

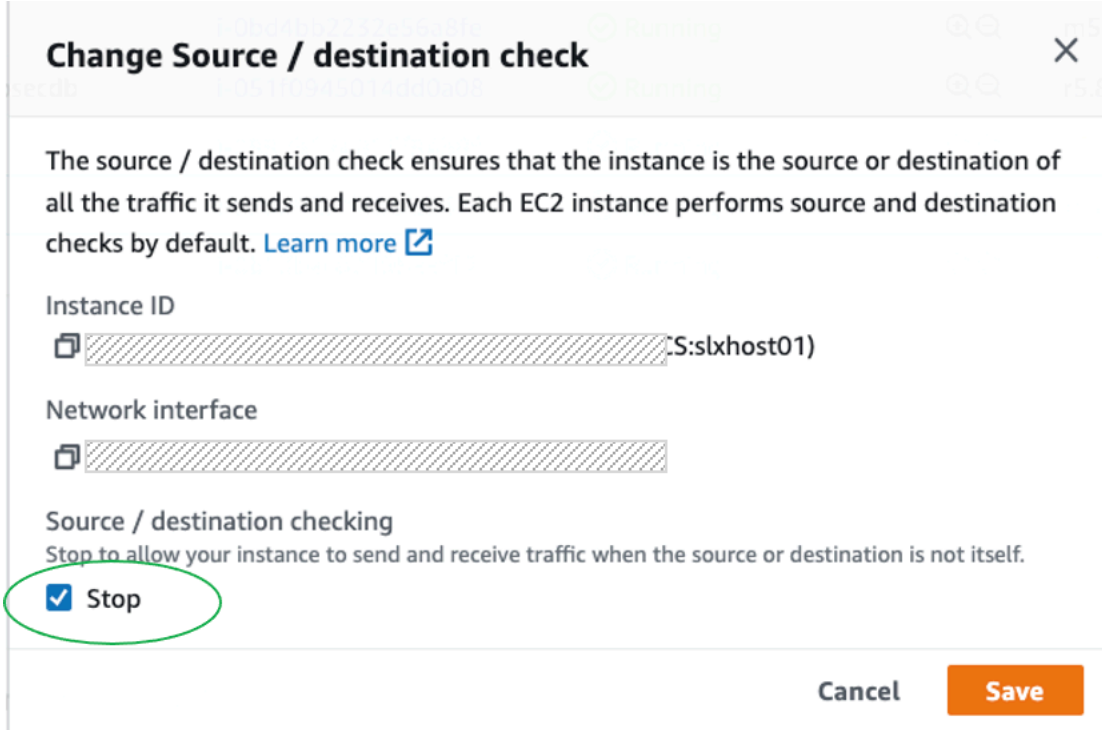
Run the following commands for both instances in the cluster.

```
aws ec2 modify-instance-attribute --instance-id <i-xxxxinstidforhost1> --no-source-dest-check
```

```
aws ec2 modify-instance-attribute --instance-id <i-xxxxinstidforhost2> --no-source-dest-check
```

AWS Management Console

Ensure that the **Stop** option is checked in https://console.aws.amazon.com/ec2/.



Change Source / destination check

The source / destination check ensures that the instance is the source or destination of all the traffic it sends and receives. Each EC2 instance performs source and destination checks by default. Learn more

Instance ID

[redacted]S:slxhost01)

Network interface

[redacted]

Source / destination checking

Stop to allow your instance to send and receive traffic when the source or destination is not itself.

☑ Stop

Review automatic recovery and stop protection

After a failure, cluster-controlled operations must be resumed in a coordinated way. This helps ensure that the cause of failure is known and addressed, and the status of the cluster is as expected. For example, verifying that there are no pending fencing actions.

This can be achieved by not enabling pacemaker to run as a service at the operating system level or by avoiding auto restarts for hardware failure.

If you want to control the restarts resulting from hardware failure, disable simplified automatic recovery and do not configure Amazon CloudWatch action-based recovery for Amazon EC2 instances that are part of a pacemaker cluster. Use the following commands on both Amazon EC2 instances in the pacemaker cluster, to disable simplified automatic recovery via AWS CLI. If making the change via AWS CLI, run the command for both Amazon EC2 instances in the cluster.

> **Note**
>
> Modifying instance maintenance options will require admin privileges not covered by the IAM instance roles defined for operations of the cluster.

```
aws ec2 modify-instance-maintenance-options --instance-id <i-xxxxinstidforhost1> --
auto-recovery disabled
```

```
aws ec2 modify-instance-maintenance-options --instance-id <i-xxxxinstidforhost2> --
auto-recovery disabled
```

To ensure that STONITH actions can be executed, you must ensure that stop protection is disabled for Amazon EC2 instances that are part of a pacemaker cluster. If the default settings have been modified, use the following commands for both instances to disable stop protection via AWS CLI.

> **Note**
>
> Modifying instance attributes will require admin privileges not covered by the IAM instance roles defined for operations of the cluster.

```
aws ec2 modify-instance-attribute --instance-id <i-xxxxinstidforhost1> --no-disable-
api-stop
```

```
aws ec2 modify-instance-attribute --instance-id <i-xxxxinstidforhost2> --no-disable-
api-stop
```

Create Amazon EC2 resource tags used by Amazon EC2 STONITH agent

Amazon EC2 STONITH agent uses AWS resource tags to identify Amazon EC2 instances. Create tag for the primary and secondary Amazon EC2 instances via AWS Management Console or AWS CLI. For more information, see Tagging your AWS resources.

Use the same tag key and the local hostname returned using the command `hostname` across instances. For example, a configuration with the values defined in Global AWS parameters would require the tags shown in the following table.

Amazon EC2	Key example	Value example
Instance 1	pacemaker	slxhost1
Instance 2	pacemaker	slxhost2

You can run the following command locally to validate the tag values and IAM permissions to describe the tags.

```
aws ec2 describe-tags --filters "Name=resource-id,Values=<instance_id>"
 "Name=key,Values=<pacemaker_tag>" --region=<region> --output=text | cut -f5
```

Operating system prerequisites

This section covers the following topics.

Topics

- Root access
- Install missing operating system packages
- Update and check operating system versions
- System logging
- Time synchronization services
- AWS CLI profile
- Pacemaker proxy settings

Root access

Verify root access on both cluster nodes. The majority of the setup commands in this document are performed with the root user. Assume that commands should be run as root unless there is an explicit call out to choose otherwise.

Install missing operating system packages

This is applicable to both cluster nodes. You must install any missing operating system packages.

The following packages and their dependencies are required for the pacemaker setup. Depending on your baseline image, for example, SLES for SAP, these packages may already be installed.

```
aws-cli
chrony
cluster-glue
corosync
crmsh
dstat
fence-agents
```

```
ha-cluster-bootstrap
iotop
pacemaker
patterns-ha-ha_sles
resource-agents
rsyslog
sap-suse-cluster-connector
sapstartsrv-resource-agents (simple-mount only)
```

We highly recommend installing the following additional packages for troubleshooting.

```
zypper-lifecycle-plugin
supportutils
yast2-support
supportutils-plugin-suse-public-cloud
supportutils-plugin-ha-sap
```

Important

Ensure that you have installed the newer version `sap-suse-cluster-connector` (**dashes**), and not the older version `sap_suse_cluster_connector` that uses underscores.

Use the following command to check packages and versions.

```
for package in aws-cli chrony cluster-glue corosync crmsh dstat fence-agents ha-
cluster-bootstrap iotop pacemaker patterns-ha-ha_sles resource-agents rsyslog sap-suse-
cluster-connector sapstartsrv-resource-agents zypper-lifecycle-plugin supportutils
 yast2-support supportutils-plugin-suse-public-cloud supportutils-plugin-ha-sap; do
echo "Checking if ${package} is installed..."
RPM_RC=$(rpm -q ${package} --quiet; echo $?)
if [ ${RPM_RC} -ne 0 ];then
echo "   ${package} is missing and needs to be installed"
fi
done
```

If a package is not installed, and you are unable to install it using `zypper`, it may be because SUSE Linux Enterprise High Availability extension is not available as a repository in your chosen image. You can verify the availability of the extension using the following command.

```
zypper repos
```

To install or update a package or packages with confirmation, use the following command.

```
zypper install <package_name(s)>
```

Update and check operating system versions

You must update and confirm versions across nodes. Apply all the latest patches to your operating system versions. This ensures that bugs are addresses and new features are available.

You can update the patches individually or use the `zypper` update. A clean reboot is recommended prior to setting up a cluster.

```
zypper update
reboot
```

Compare the operating system package versions on the two cluster nodes and ensure that the versions match on both nodes.

System logging

This is applicable to both cluster nodes. We recommend using the `rsyslogd` daemon for logging. It is the default configuration in the cluster. Verify that the `rsyslog` package is installed on both cluster nodes.

`logd` is a subsystem to log additional information coming from the STONITH agent.

```
systemctl enable --now logd
systemctl status logd
```

Time synchronization services

This is applicable to both cluster nodes. Time synchronization is important for cluster operation. Ensure that `chrony` rpm is installed, and configure appropriate time servers in the configuration file.

You can use Amazon Time Sync Service that is available on any instance running in a VPC. It does not require internet access. To ensure consistency in the handling of leap seconds, don't mix Amazon Time Sync Service with any other `ntp` time sync servers or pools.

Create or check the `/etc/chrony.d/ec2.conf` file to define the server.

```
#Amazon EC2 time source config
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

Start the `chronyd.service`, using the following command.

```
systemctl enable --now chronyd.service
systemctl status chronyd
```

For more information, see Set the time for your Linux instance.

AWS CLI profile

This is applicable to both cluster nodes. The cluster resource agents use AWS Command Line Interface (AWS CLI). You need to create an AWS CLI profile for the root account on both instances.

You can either edit the config file at `/root/.aws` manually or by using aws configure AWS CLI command.

You can skip providing the information for the access and secret access keys. The permissions are provided through IAM roles attached to Amazon EC2 instances.

```
aws configure
{aws} Access Key ID [None]:
{aws} Secret Access Key [None]:
Default region name [None]: <region_id>
Default output format [None]:
```

The profile name is configurable. The name chosen in this example is **cluster** – it is used in Create Amazon EC2 resource tags used by Amazon EC2 STONITH agent. The AWS Region must be the default AWS Region of the instance.

```
aws configure --profile <cluster>
{aws} Access Key ID [None]:
{aws} Secret Access Key [None]:
Default region name [None]: <region_id>
Default output format [None]:
```

Pacemaker proxy settings

This is applicable to both cluster nodes. If your Amazon EC2 instance has been configured to access the internet and/or AWS Cloud through proxy servers, then you need to replicate the settings in the pacemaker configuration. For more information, see Use an HTTP proxy.

Add the following lines to `/etc/sysconfig/pacemaker`.

```
http_proxy=http://<proxyhost>:<proxyport>
https_proxy= http://<proxyhost>:<proxyport>
no_proxy=127.0.0.1,localhost,169.254.169.254,fd00:ec2::254
```

Modify `proxyhost` and `proxyport` to match your settings. Ensure that you exempt the address used to access the instance metadata. Configure `no_proxy` to include the IP address of the instance metadata service – **`169.254.169.254`** (IPV4) and **`fd00:ec2::254`** (IPV6). This address does not vary.

IP and hostname resolution prerequisites

This section covers the following topics.

Topics

- Primary and secondary IP addresses
- Add initial VPC route table entries for overlay IPs
- Add overlay IPs to host IP configuration
- Hostname resolution

Primary and secondary IP addresses

This is applicable to both cluster nodes. We recommend defining a redundant communication channel (a second ring) in `corosync` for SUSE clusters. The cluster nodes can use the second ring to communicate in case of underlying network disruptions.

Create a redundant communication channel by adding a secondary IP address on both nodes.

Add a secondary IP address on both nodes. These IPs are only used in cluster configurations. They provide the same fault tolerance as a secondary Elastic Network Interface (ENI). For more information, see Assign a secondary private IPv4 address.

On correct configuration, the following command returns two IPs from the same subnet on both, the primary and secondary node.

```
ip -o -f inet addr show eth0 | awk -F " |/" '{print $7}'
```

These IP addresses are required for `ring0_addr` and `ring1_addr` in `corosync.conf`.

Add initial VPC route table entries for overlay IPs

You need to add initial route table entries for overlay IPs. For more information on overlay IP, see Overlay IP.

Add entries to the VPC route table or tables associated with the subnets of your Amazon EC2 instance for the cluster. The entries for destination (overlay IP CIDR) and target (Amazon EC2 instance or ENI) must be added manually for ASCS and ERS. This ensures that the cluster resource has a route to modify. It also supports the install of SAP using the virtual names associated with the overlay IP before the configuration of the cluster.

Modify or add a route to a route table using AWS Management Console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Route Tables**, and select the route table associated with the subnets where your instances have been deployed.
3. Choose **Actions**, **Edit routes**.
4. To add a route, choose **Add route**. You must choose **Add route** twice to add two routes, one for ASCS and another for ERS.

   a. Add your chosen overlay IP address CIDR and the instance ID of your primary instance for ASCS. See the following table for an **example**.

Destination	172.16.30.5/32
Target	i-xxxxinstidforhost1

   b. Add your chosen overlay IP address CIDR and the instance ID of your secondary instance for ERS. See the following table for an **example**.

Destination	172.16.30.6/32

Target	i-xxxxinstidforhost2

5. Choose **Save changes**.

   You route table now has two entries, each for ASCS and ERS, in addition to the standard routes. The selected instances IDs resolve to the corresponding primary Elastic Network Interface (ENI).

The preceding steps can also be performed programmatically. We suggest performing the steps using administrative privileges, instead of instance-based privileges to preserve least privilege. CreateRoute API isn't necessary for ongoing operations.

Run the following command as a dry run on both nodes to confirm that the instances have the necessary permissions.

```
aws ec2 replace-route --route-table-id <rtb-xxxxxroutetable1> --destination-cidr-
block <172.16.30.5/32> --instance-id <i-xxxxinstidforhost1> --dry-run --profile
 <aws_cli_cluster_profile>
```

Add overlay IPs to host IP configuration

You must configure the overlay IP as an additional IP address on the standard interface to enable SAP install. This action is managed by the cluster IP resource. However, to install SAP using the correct IP addresses prior to having the cluster configuration in place, you need to add these entries manually.

If you need to reboot the instance during setup, the assignment is lost, and must be re-added.

See the following **examples**. You must update the commands with your chosen IP addresses.

On EC2 instance 1, where you are installing ASCS, add the overlay IP allocated for ASCS.

```
ip addr add <172.16.30.5/32> dev eth0
```

On EC2 instance 2, where you are installing ERS, add the overlay IP allocated for ERS.

```
ip addr add <172.16.30.6/32> dev eth0
```

Hostname resolution

This is applicable to both cluster nodes. You must ensure that both instances can resolve all hostnames in use. Add the hostnames for cluster nodes to `/etc/hosts` file on both cluster nodes.

This ensures that hostnames for cluster nodes can be resolved even in case of DNS issues. See the following example.

```
cat /etc/hosts
<10.1.10.1 slxhost01.example.com slxhost01>
<10.1.20.1 slxhost02.example.com slxhost02>
<172.16.30.5 slxascs.example.com slxascs>
<172.16.30.6 slxers.example.com slxers>
```

In this example, the secondary IPs used for the second cluster ring are not mentioned. They are only used in the cluster configuration. You can allocate virtual hostnames for administration and identification purposes.

> ⚠ **Important**
>
> The overlay IP is out of VPC range, and cannot be reached from locations not associated with the route table, including on-premises.

File system prerequisites

This section covers the following topics.

Topics

- Shared file systems
- Create file systems
- Update /etc/fstab
- Temporarily mount ASCS and ERS directories for installation (classic only)

Shared file systems

Amazon Elastic File System and Amazon FSx for NetApp ONTAP both support SAP NetWeaver file systems on Linux. For more information, see Getting started with Amazon Elastic File System and Getting started with Amazon FSx for NetApp ONTAP. You can evaluate both of these options, and select one based on resilience, performance, and cost.

We recommend sharing a single Amazon EFS or FSx for ONTAP file system across multiple SIDs within an account.

The file system's DNS name is the simplest mounting option. The DNS automatically resolves to the mount target's IP address in the Availability Zone of the connecting Amazon EC2 instance. You can also create an alias to help with identifying the purpose of the shared file system. We have used `<nfs.fqdn>` in this document. The following are some examples.

- file-system-id.efs.aws-region.amazonaws.com
- svm-id.fs-id.fsx.aws-region.amazonaws.com
- qas_sapmnt_share.example.com

Note

Review the `enableDnsHostnames` and `enableDnsSupport` DNS attributes for your VPC. For more information, see View and update DNS attributes for your VPC.

Create file systems

The following shared file systems are covered in this document.

Unique NFS Location (example)	File system location
SLX_sapmnt	/sapmnt/SLX
SLX_ASCS00	/usr/sap/SLX/ASCS00
SLX_ERS10	/usr/sap/SLX/ERS10

For more information, see SAP Documentation – SAP System Directories on UNIX.

The following options can differ depending on how you architect and operate your systems.

- When using the simple mount architecture, you can share at the `/usr/sap/<SID>` level. There is no requirement to manage these file systems with the cluster. In this document, these file systems are separate. This is to simplify the migration, and to ensure that the recommendation to run app server executables from a local copy is followed in case you are co-hosting the ASCS/ERS with an application server.
- `/usr/sap/trans` is not listed as a required file system for ASCS. You can add this to your setup.

- A shared home directory has not been included. This enables you to log on locally as the `<sid>adm` user, in the event of NFS issues. A shared home directory can be the suitable option if your administrators have root access.

Using the NFS ID created in the previous step, temporarily mount the root directory of the NFS with the following command. `/mnt` is available by default; it can also be substituted with another temporary location.

```
mount <nfs.fqdn>.amazonaws.com:/ /mnt
```

Create the directories using the following command.

```
mkdir -p /mnt/<SLX_sapmnt>
mkdir -p /mnt/<SLX_ASCS00>
mkdir -p /mnt/<SLX_ERS10>
```

Change the ownership or permissions to ensure that the install as `<sid>adm` can write to the directories. If `<sid>adm` is going to be created by the installer, temporarily change the permissions to 777, as seen in the following command.

```
chmod 777 /mnt/<SLX_sapmnt> /mnt/<SLX_ASCS00> /mnt/<SLX_ERS10>
```

Unmount the temporary mount using the following command.

```
umount /mnt
```

Update `/etc/fstab`

This is applicable to both cluster nodes. `/etc/fstab` is a configuration table containing the details required for mounting and unmounting file systems to a host.

Create the directories for the required mount points (permanent or cluster controlled), using the following commands.

```
mkdir /sapmnt
mkdir /usr/sap/<SLX>/ASCS
mkdir /usr/sap/<SLX>/ERS2
```

Add the file systems not managed by the cluster to `/etc/fstab`.

For both simple-mount and classic architectures, prepare and append an entry for the `sapmnt` file system to `/etc/fstab`.

```
<nfs.fqdn>/<SLX_sapmnt>   /sapmnt    nfs
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 0 0
```

Simple-mount only – prepare and append entries for the ASCS and ERS file systems to `/etc/fstab`.

```
<nfs.fqdn>:/<SLX_ASCS00>   /usr/sap/<SLX>/ASCS<00>    nfs
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 0 0

<nfs.fqdn>:/<SLX_ERS10>   /usr/sap/<SLX>/ERS<10>   nfs
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 0 0
```

> ⚠ **Important**
>
> Review the mount options to ensure that they match with your operating system, NFS file system type, and latest recommendations from SAP.

Use the following command to mount the file systems defined in `/etc/fstab`.

```
mount -a
```

Use the following command to check that the required file systems are available.

```
df -h
```

Temporarily mount ASCS and ERS directories for installation (classic only)

This is only applicable to the classic architecture. Simple-mount architecture has these directories permanently available in `/etc/fstab`.

You must temporarily mount ASCS and ERS directories for installation.

Use the following command on the instance where you to plan to install ASCS.

```
mount <nfs.fqdn>:/<SLX_ASCS00>  /usr/sap/<SLX>/ASCS
```

Use the following command on the instance where you to plan to install ERS.

```
mount <nfs.fqdn>:/<SLX_ERS10>  /usr/sap/<SLX>/ERS
```

Shared VPC – *optional*

Amazon VPC sharing enables you to share subnets with other AWS accounts within the same AWS Organizations. Amazon EC2 instances can be deployed using the subnets of the shared Amazon VPC.

In the pacemaker cluster, the `aws-vpc-move-ip` resource agent has been enhanced to support a shared VPC setup while maintaining backward compatibility with previous existing features.

The following checks and changes are required. We refer to the AWS account that owns Amazon VPC as the sharing VPC account, and to the consumer account where the cluster nodes are going to be deployed as the cluster account.

This section covers the following topics.

Topics

- Minimum version requirements
- IAM roles and policies
- Shared VPC cluster resources

Minimum version requirements

The latest version of the `aws-vpc-move-ip` agent shipped with SLES15 SP3 supports the shared VPC setup by default. The following are the minimum version required to support a shared VPC Setup:

- SLES 12 SP5 - resource-agents-4.3.018.a7fb5035-3.79.1.x86_64
- SLES 15 SP2 - resource-agents-4.4.0+git57.70549516-3.30.1.x86_64
- SLES 15 SP3 - resource-agents-4.8.0+git30.d0077df0-8.5.1

IAM roles and policies

Using the overlay IP agent with a shared Amazon VPC requires a different set of IAM permissions to be granted on both AWS accounts (sharing VPC account and cluster account).

Sharing VPC account

In sharing VPC account, create an IAM role to delegate permissions to the EC2 instances that will be part of the cluster. During the IAM Role creation, select "Another AWS account" as the type of trusted entity, and enter the AWS account ID where the EC2 instances will be deployed/running from.

After the IAM role has been created, create the following IAM policy on the sharing VPC account, and attach it to an IAM role. Add or remove route table entries as needed.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "ec2:ReplaceRoute",
      "Resource": [
        "arn:aws:ec2:<region>:<sharing_vpc_account_id>:route_table/<rtb_id_1>",
        "arn:aws:ec2:<region>:<sharing_vpc_account_id>:route_table/<rtb_id_2>"
      ]
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "ec2:DescribeRouteTables",
      "Resource": "*"
    }
  ]
}
```

Next, edit move to the "Trust relationships" tab in the IAM role, and ensure that the AWS account you entered while creating the role has been correctly added.

Cluster account

In cluster account, create the following IAM policy, and attach it to an IAM role. This is the IAM Role that is going to be attached to the EC2 instances.

AWS STS policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::<sharing_vpc_account_id>:role/<sharing _vpc-account-
cluster-role>"
    }
  ]
}
```

STONITH policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": [
        "arn:aws:ec2:<region>:<cluster_account_id>:instance/<instance_id_1>",
        "arn:aws:ec2:<region>:<cluster_account_id>:instance/<instance_id_2>"
      ]
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    }
  ]
}
```

Shared VPC cluster resources

The cluster resource agent `aws-vpc-move-ip` also uses a different configuration syntax. When configuring the `aws-vpc-move-ip` resource agent, the following new parameters must be used:

- lookup_type=NetworkInterfaceId
- routing_table_role="arn:aws:iam::<account_id>:role/<VPC-Account-Cluster-Role>"

The following IP Resource for ASCS needs to be created.

```
crm configure primitive rsc_ip_<SLX>_ASCS<00> ocf:heartbeat:aws-vpc-
move-ip params ip=<172.16.30.5> routing_table=<rtb-xxxxxroutetable1>
 interface=eth0 profile=<cluster> lookup_type=NetworkInterfaceId
 routing_table_role="arn:aws:iam::<sharing_vpc_account_id>:role/
<sharing_vpc_account_cluster_role>" op start interval=0 timeout=180s op stop interval=0
 timeout=180s op monitor interval=20s timeout=40s
```

The following IP Resource for ERS needs to be created.

```
crm configure primitive rsc_ip_<SLX>_ERS<10> ocf:heartbeat:aws-vpc-
move-ip params ip=<172.16.30.6> routing_table=<rtb-xxxxxroutetable1>
 interface=eth0 profile=<cluster> lookup_type=NetworkInterfaceId
 routing_table_role="arn:aws:iam::<sharing_vpc_account_id>:role/
<sharing_vpc_account_cluster_role>" op start interval=0 timeout=180s op stop interval=0
 timeout=180s op monitor interval=20s timeout=40s
```

SAP and cluster setup

This section covers the following topics.

Topics

Install SAP

The following topics provide information about installing SAP on AWS Cloud in a highly available cluster. Review SAP Documentation for more details.

Topics

- Use SWPM with high availability
- Install SAP instances
- Kernel upgrade and ENSA2 – optional
- Check SAP host agent version

Use SWPM with high availability

Before running SAP Software Provisioning Manager (SWPM), ensure that the following prerequisites are met.

- If the operating system groups for SAP are pre-defined, ensure that the user identifier (UID) and group identifier values for `<sid>adm` and `sapsys` are consistent across both instances.
- You have downloaded the most recent version of Software Provisioning Manager for your SAP version. For more information, see SAP Documentation Software Provisioning Manager.
- Ensure that routes, overlay IPs, and virtual host names are mapped to both instances. This is to ensure that the virtual hostname for ASCS is available on instance 1, and the virtual hostname for ERS is available on instance 2. For more information, see IP and hostname resolution prerequisites.
- Ensure that shared file systems are available, either in `/etc/fstab` or using the mount command. For more information, see File system prerequisites.

Install SAP instances

The commands in this section use the example values provided in Define reference parameters for setup.

Install ASCS instance on `slxhost01` with virtual hostname `slxascs`, using the high availability option of Software Provisioning Manager (SWPM) tool. You can use the `SAPINST_USE_HOSTNAME` parameter to install SAP using a virtual hostname.

```
<swpm location>/sapinst SAPINST_USE_HOSTNAME=<slxascs>
```

Install ERS instance on `slxhost02` with virtual hostname `slxers`, using the high availability option of Software Provisioning Manager (SWPM) tool. You can use the `SAPINST_USE_HOSTNAME` parameter to install SAP using a virtual hostname.

```
<swpm location>/sapinst SAPINST_USE_HOSTNAME=<slxers>
```

Once installation is complete, install and configure the database and SAP Primary Application Server (PAS). Optionally, you can also install and configure Additional Application Server (AAS). For more details on installing SAP NetWeaver, refer SAP Help Portal.

For additional information on unattended installation options, see SAP Note 2230669 – System Provisioning Using an Input Parameter File (requires SAP portal access).

Kernel upgrade and ENSA2 – *optional*

As of AS ABAP Release 7.53 (ABAP Platform 1809), the new Standalone Enqueue Server 2 (ENSA2) is installed by default. ENSA2 replaces the previous version – ENSA1.

If you have an older version of SAP NetWeaver, consider following the SAP guidance to upgrade the kernel and update the Enqueue Server configuration. An upgrade will allow you to take advantage of the features available in the latest version. For more information, see the following SAP Notes (require SAP portal access).

- SAP Note 2630416 – Support for Standalone Enqueue Server 2
- SAP Note 2711036 – Usage of the Standalone Enqueue Server 2 in an HA Environment

Check SAP host agent version

This is applicable to both cluster nodes. The SAP host agent is used for system instance control and monitoring. This agent is used by SAP cluster resource agents and hooks. It is recommended that you have the latest version installed on both instances. For more details, see SAP Note 2219592 – Upgrade Strategy of SAP Host Agent.

Use the following command to check the version of the host agent.

```
/usr/sap/hostctrl/exe/saphostexec -version
```

Modify SAP control operations for cluster use

This section covers the following topics.

Topics

- Add sidadm to haclient group
- Modify SAP profiles for start operations and cluster hook
- Modify sapservices
- Align and disable SAP auto start services for systemd
- Enable sapping/sappong systemd services (simple-mount only)

Add `sidadm` to `haclient` group

This is applicable to both cluster nodes. An `haclient` operating system group is created when the cluster connector package is installed. Adding the `sidadm` user to this group ensures that your cluster has necessary access. Run the following command as root.

```
usermod -a -G haclient <slx>adm
```

Modify SAP profiles for start operations and cluster hook

This action ensures that there is compatibility between SAP start framework and cluster actions. Modify SAP profiles to change the start behavior of the SAP instance and processes. Ensure that `sapcontrol` is aware that the system is being managed by a pacemaker cluster.

The following changes must be made in the instance profiles for ASCS and ERS. These profiles are created during install, and are located at `/usr/sap/<SID>/SYS/profile/`.

- ASCS profile example – `/usr/sap/SLX/SYS/profile/SLX_ASCS00_slxascs`
- ERS profile example – `/usr/sap/SLX/SYS/profile/SLX_ERS10_slxers`

1. **Program or process start behavior** – In case of failure, processes must be restarted. Determining where the process starts and in what order needs to be controlled by the cluster, and not SAP start framework behavior defined in the profiles. Your locks can be lost if this parameter is not changed.

   Example

   ENSA1

   ASCS

   ```
   #For ENSA1 (_EN)
   ```

```
#Changing Restart to Start for Cluster compatibility
#Old value: Restart_Program_XX = local $(_EN) pf=$(_PF)

Start_Program_XX = local $(_EN) pf=$(_PF)
```

ERS

```
#For ENSA1 (_ER)
#Changing Restart to Start for Cluster compatibility
#Old value: Restart_Program_XX = local $(_ER) pf=$(_PFL)NR=$(SCSID)

Start_Program_XX = local $(_ER) pf=$(_PFL) NR=$(SCSID)
```

XX *indicates the start-up order. This value may be different in your install; retain the unchanged value.*

ENSA2

ASCS

```
#For ENSA2 (_ENQ)
#Changing Restart to Start for Cluster compatibility
#Old value: Restart_Program_XX = local $(_ENQ) pf=$(_PF)

Start_Program_XX = local $(_ENQ) pf=$(_PF)
```

ERS

```
#For ENSA2 (_ENQR)
#Changing Restart to Start for Cluster compatibility
#Old value: Restart_Program_XX = local $(_ENQR) pf=$(_PFL)NR=$(SCSID)

Start_Program_XX = local $(_ENQR) pf=$(_PFL) NR=$(SCSID)
```

XX *indicates the start order. This value may be different in your install; retain the unchanged value.*

2. **Disable instance auto start in both profiles –** When an instance restarts, SAP start framework should not start ASCS and ERS automatically. Add the following parameter on both profiles to prevent an auto start.

```
Autostart = 0
```

3. **Add cluster connector details in both profilesP –** The connector integrates the SAP start and control frameworks of SAP NetWeaver with SUSE cluster to assist with maintenance and awareness of state. Add the following parameters on both profiles.

   ```
   #Added for Cluster Connectivity
   service/halib = $(DIR_CT_RUN)/saphascriptco.so
   service/halib_cluster_connector = /usr/bin/sap_suse_cluster_connector
   ```

   > ⚠ **Important**
   >
   > RPM package `sap-suse-cluster-connector` has *dashes*. The executable `/usr/bin/sap_suse_cluster_connector` available after installation has *underscores*. Ensure that the correct name, that is executable `/usr/bin/sap_suse_cluster_connector`, is used in both profiles.

4. **Restart services –** Restart SAP services for ASCS and ERS to ensure that the preceding settings take effect. Adjust the system number to match the service.

   ASCS

   ```
   /usr/sap/hostctrl/exe/sapcontrol -nr <00> -function RestartService
   ```

   ERS

   ```
   /usr/sap/hostctrl/exe/sapcontrol -nr <10> -function RestartService
   ```

5. **Check integration using `sapcontrol` –** `sapcontrol` includes two functions: `HACheckConfig` and `HACheckFailoverConfig`. These functions can be used to check configuration, including awareness of the cluster connector.

   ASCS

   ```
   /usr/sap/hostctrl/exe/sapcontrol -nr <00> -function HACheckFailoverConfig
   /usr/sap/hostctrl/exe/sapcontrol -nr <00> -function HACheckConfig
   ```

Modify `sapservices`

This is applicable to both cluster nodes. In older versions of SLES and SAP kernel, the `systemV init` service `sapinit` is responsible for starting SAP host agent and all `sapstartsrv` processes listed in `/usr/sap/sapservices`. In newer versions, native integration is available between `systemd` and SAP services. For more details, see the following SAP Notes (require SAP portal access).

- SAP Note 3139184 – Linux: systemd integration for sapstartsrv and SAP Host Agent
- SAP Note 3115048 – sapstartsrv with native Linux systemd support

Review if `systemV` or `systemd` integration is in place, and is consistent for ASCS and ERS by checking the `/usr/sap/services` file on both nodes.

```
cat /usr/sap/sapservices
```

See the following table for more details.

Example

systemV

See the following example ASCS entry for older version with `systemV` integration.

```
LD_LIBRARY_PATH=/usr/sap/<SLX>/ASCS<00>/exe:$LD_LIBRARY_PATH;export
 LD_LIBRARY_PATH;/usr/sap/<SLX>/ASCS<00>/exe/sapstartsrv pf=/usr/sap/<SLX>/SYS/
profile/<SLX>_ASCS<00>_<slxascs> -D -u <slxadm>
```

To ensure that SAP instance can be managed by the cluster and also manually during planned maintenance activities, add the missing entries for ASCS and ERS `sapstartsrv` service in `/usr/sap/sapservices` file on both cluster nodes (ASCS and ERS host). Copy the missing entry from both hosts. Post-modifications, the `/usr/sap/sapservices` file looks as follows on both hosts.

```
#!/bin/sh
LD_LIBRARY_PATH=/usr/sap/<SLX>/ASCS<00>/exe:$LD_LIBRARY_PATH; export
 LD_LIBRARY_PATH; /usr/sap/<SLX>/ASCS<00>/exe/sapstartsrv pf=/usr/sap/<SLX>/SYS/
profile/<SLX>_ASCS<00>_<slxascs> -D -u <slxadm>
```

```
LD_LIBRARY_PATH=/usr/sap/<SLX>/ERS<10>/exe:$LD_LIBRARY_PATH; export
 LD_LIBRARY_PATH; /usr/sap/<SLX>/ERS<10>/exe/sapstartsrv pf=/usr/sap/<SLX>/SYS/
profile/<SLX>_ERS<10>_<slxers> -D -u <slxadm>
```

systemd

See the following example ASCS entry for newer version with native `systemd` integration.

```
systemctl --no-ask-password start <SAPSLX>_<00> # sapstartsrv pf=/usr/sap/<SLX>/SYS/
profile/SLX_ASCS<00>_<slxascs>
```

To ensure that SAP instance can be managed by the cluster and also manually during planned maintenance activities, add the missing entries for ASCS and ERS `sapstartsrv` service in `/usr/sap/sapservices` file on both cluster nodes (ASCS and ERS host). Copy the missing entry from both hosts. Post-modifications, the `/usr/sap/sapservices` file looks as follows on both hosts.

```
#!/bin/sh
systemctl --no-ask-password start <SAPSLX_00> # sapstartsrv pf=/usr/sap/<SLX>/SYS/
profile/SLX_ASCS<00>_<slxascs>

systemctl --no-ask-password start <SAPSLX_10> # sapstartsrv pf=/usr/sap/<SLX>/SYS/
profile/SLX_ERS<10>_<slxers>
```

Align and disable SAP auto start services for `systemd`

This is applicable to both cluster nodes. For `systemd`, ensure SAP auto start services are aligned and disabled across nodes. If the installed version supports native integration with `systemd`, you must create services for ASCS and ERS on both nodes. This ensures that if you have revert to manual operations, there is no association and that both nodes are configured in the same manner.

You must disable auto start services to enable the cluster to manage stop/start.

ASCS

Register the missing ERS service on the node where you have installed ASCS.

Temporarily mount the ERS directory (classic only).

```
mount <nfs.fqdn>:/<SLX>_ERS<10>  /usr/sap/<SLX>/ERS10
```

Register the ERS service.

```
export LD_LIBRARY_PATH=/usr/sap/<SLX>/ERS10xe
/usr/sap/<SLX>/ERS10/exe/sapstartsrv pf=/usr/sap/<SLX>/SYS/profile/SLX_ERS10_slxers -
reg

systemctl start SAPSLX_10
```

Check the existence and state of SAP services.

```
systemctl list-unit-files SAP*
UNIT FILESTATE VENDOR PRESET
SAPSLX_00.service disabled disabled
SAPSLX_10.service disabled disabled
SAP.slicestatic -
3 unit files listed.
```

If the state is not disabled, run the following command to disable `sapservices` integration for `SAP<SID>_<ascs_sys_nr>` and `SAP<SID>_<ers_sys_nr>` on both nodes.

> ⚠ **Important**
>
> Stopping these services also stops the associated SAP instances.

```
systemctl stop SAP<SLX>_<00>.service
systemctl disable SAP<SLX>_<00>.service
systemctl stop SAP<SLX>_<10>.service
systemctl disable SAP<SLX>_<10>.service
```

Unmount the ERS directory (classic only).

```
umount /usr/sap/<SLX>/ERS
```

ERS

Register the missing ASCS service on the node where you have installed ERS.

Temporarily mount the ASCS directory (classic only).

```
mount <nfs.fqdn>:/SLX_ASCS00  /usr/sap/<SLX>/ASCS
```

Register the ASCS service.

```
export LD_LIBRARY_PATH=/usr/sap/<SLX>/ASCS<00>/exe
/usr/sap/<SLX>/ASCS<00>/exe/sapstartsrv pf=/usr/sap/<SLX>/SYS/profile/
SLX_ASCS<00>_<slxascs> -reg
systemctl start SAP<SLX>_
```

Check the existence and state of SAP services.

```
systemctl list-unit-files SAP*
UNIT FILESTATE VENDOR PRESET
SAP<SLX>_<00>.service disabled disabled
SAP<SLX>_<10>.service disabled disabled
SAP.slicestatic -
3 unit files listed.
```

If the state is not disabled, run the following command to disable `sapservices` integration for `SAP<SID>_<ascs_sys_nr>` and `SAP<SID>_<ers_sys_nr>` on both nodes.

Important

Stopping these services also stops the associated SAP instances.

```
systemctl stop SAP<SLX>_<00>.service
systemctl disable SAP<SLX>_<00>.service
systemctl stop SAP<SLX>_<10>.service
systemctl disable SAP<SLX>_<10>.service
```

Unmount the ASCS directory (classic only).

```
umount /usr/sap/<SLX>/ASCS
```

For more details, see SAP Note 3139184 – Linux: systemd integration for sapstartsrv and SAP Host Agent.

Enable `sapping`/`sappong` systemd services (simple-mount only)

In simple-mount architecture, the `sapstartsrv` resource is managed by the cluster. `sapstartsrv` should not be started by `sapinit` boot script during cluster nodes startup. The new services – `sapping` and `sappong` are used to mask and unmask the `/usr/sap/sapservices` file to meet this requirement.

These services are introduced by the `sapstartsrv` resource agent (located in the package `sapstartsrv-resource-agents`), and must exist in disabled state. If they do not exist, use the following command to check that you have installed the package.

```
zypper info sapstartsrv-resource-agents
```

Run the following command to enable `sapping` and `sappong` services on both nodes.

```
systemctl enable sapping
systemctl enable sappong
```

Cluster prerequisites

This section covers the following topics.

Topics

- Configure systemd for resource dependencies
- Update the hacluster password
- Setup passwordless authentication between nodes
- Create an authentication key for corosync

Configure `systemd` for resource dependencies

This is applicable to both cluster nodes. Some failure scenarios, such as an accidental shutdown of an Amazon EC2 instance, can result in unexpected fencing actions. This is caused by pacemaker dependencies that are not directly associated with cluster resources and constraints, but instead are a dependency for the pacemaker service.

If `systemd` is configured, create a config file that defines a dependency between pacemaker and SAP services.

```
mkdir -p /etc/systemd/system/resource-agents-deps.target.d/
cd /etc/systemd/system/resource-agents-deps.target.d/


cat > sap_systemd_<slx>.conf <<_EOF
[Unit]
Requires=sapinit.service
After=sapinit.service
After=SAP<SLX>_<00>.service
After=SAP<SLX>_<10>.service
_EOF


systemctl daemon-reload
```

Update the `hacluster` password

This is applicable to both cluster nodes. Change the password of the operating system user `hacluster` using the following command.

```
passwd hacluster
```

Setup passwordless authentication between nodes

For a more comprehensive and easily consumable view of cluster activity, SUSE provides additional reporting tools. Many of these tools require access to both nodes without entering a password. SUSE recommends performing this setup for root user. For more details, see *Configuration to collect cluster report as root with root SSH access between cluster nodes* section in SUSE Documentation https://www.suse.com/support/kb/doc/?id=000017501#:~:text=The%20hb_report%20utility%20(on%20newer,an%20incident%20to%20be%20investigated.[Usage of hb_report for SLES HAE].

Create an authentication key for `corosync`

If you want to configure `corosync` to use cryptographic techniques for ensuring authenticity and privacy of the messages, you need to generate a private key. The executable `corosync-keygen` creates this key and writes it to `/etc/corosync/authkey`.

Use the following command on Node 1 as root.

```
corosync-keygen
```

Use `scp` or a temporary shared NFS location to copy an identical file on the second node at the same location. For example, on `slxhost01`.

```
scp -p /etc/corosync/authkey root@<slxhost02>:/etc/corosync
```

Create cluster and node associations

This section covers the following topics.

Topics

- Stop services for initial configuration
- File modifications and key values
- Sample corosync.conf file

Stop services for initial configuration

This is applicable to both cluster nodes. The cluster service `pacemaker` must be in a stopped state when performing cluster configuration.

Run the following command to check if `pacemaker` is running.

```
systemctl status pacemaker
```

Run the following command to stop `pacemaker`.

```
systemctl stop pacemaker
```

File modifications and key values

`corosync.conf` is the configuration file for the `corosync` executable. Copy the contents of the Sample corosync.conf file to `/etc/corosync/corosync.conf` on both nodes.

Ensure the following when copying the file.

- Ensure that the node list IP addresses match the primary and secondary IPs on each host (not the overlay IP)
- Ensure that the file is same on both nodes, with the exception of `bindnetaddr` that should match the relevant local primary IP address on each node.
- Ensure that the token value is set to 30000. This timeout specifies the time taken in milliseconds until a token loss is declared after not receiving a token. This is important for the stability of the cluster.

Sample `corosync.conf` file

The following is a sample `corosync.conf` file.

Ensure that the file is same on both nodes, with the exception of `bindnetaddr` that should match the relevant local primary IP address on each node.

```
#Read the corosync.conf.5 manual page
totem {
  version: 2
  rrp_mode: passive
  token: 30000
  consensus: 36000
  token_retransmits_before_loss_const: 10
  max_messages: 20
  crypto_cipher: aes256
  crypto_hash: sha1
  clear_node_high_bit: yes
  interface {
    ringnumber: 0
    bindnetaddr: <local_ip>
    mcastport: 5405
    ttl: 1
 }
  transport: udpu
}
 logging {
      fileline: off
      to_logfile: yes
      to_syslog: yes
      logfile: /var/log/cluster/corosync.log
      debug: off
      timestamp: on
      logger_subsys {
         subsys: QUORUM
         debug: off
     }
}
nodelist {
  node {
  ring0_addr: <primary_host_ip>
  ring1_addr: <primary_host_additional_ip>
  nodeid: 1
  }
  node {
  ring0_addr: <secondary_host_ip>
  ring1_addr: <secondary_host_additional_ip>
  nodeid: 2
  }
}

quorum {
  # Enable and configure quorum subsystem (default: off)
  # see also corosync.conf.5 and votequorum.5
  provider: corosync_votequorum
  expected_votes: 2
  two_node: 1
}
```

The following table displays example substitutions for IP addresses using the sample IP addresses provided in this document. The <local_ip> configuration differs between hosts.

IP address type	Primary host	Secondary host
<local_ip>	**10.1.10.1**	**10.1.20.1**
<primary_host_ip>	10.1.10.1	10.1.10.1
<primary_host_additional_ip>	10.1.10.2	10.1.10.2
<secondary_host_ip>	10.1.20.1	10.1.20.1
<secondary_host_additional_ip>	10.1.20.2	10.1.20.2

Cluster configuration

This section covers the following topics.

Topics

Cluster resources

This section covers the following topics.

Topics

Enable and start the cluster

This is applicable to both cluster nodes. Run the following command to enable and start the `pacemaker` cluster service on both nodes.

```
systemctl enable --now pacemaker
```

or

```
systemctl start pacemaker
```

By enabling the `pacemaker` service, the server automatically joins the cluster after a reboot. This ensures that your system is protected. Alternatively, you can start the `pacemaker` service manually on boot. You can then investigate the cause of failure. However, this is generally not required for SAP NetWeaver ASCS cluster.

Run the following command to check the status of the `pacemaker` service.

```
systemctl status pacemaker
# pacemaker.service - Pacemaker High Availability Cluster Manager
     Loaded: loaded (/usr/lib/systemd/system/pacemaker.service; <enabled>; vendor
 preset: disabled)
     Active: <active (running)> since Tue XXXX-XX-XX XX:XX:XX XXX; XXh ago
       Docs: man:pacemakerd
             https://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/2.0/html-single/
Pacemaker_Explained/index.html
   Main PID: 1899 (pacemakerd)
Enable cluster service (optional)
```

Check cluster status

Once the cluster service `pacemaker` is started, check the cluster status with `crm mon` command, as shown in the following example.

```
crm_mon -1
Cluster Summary:
  * Stack: corosync
  * Current DC: <slxhost01> (version 2.0.xxxxxxxxxxx) - partition with quorum
  * Last updated:
  * Last change:  by hacluster via crmd on slxhost01
  * 2 nodes configured
  * 0 resource instances configured

Node List:
  * Online: [ <slxhost01> <slxhost02> ]
```

```
Active Resources:
  * No active resources
```

The primary (`slxhost01`) and secondary (`slxhost02`) must show up as online.

You can find the ring status and the associated IP address of the cluster with `corosync-cfgtool` command, as shown in the following example.

```
corosync-cfgtool -s
Printing ring status.
Local node ID 1
RING ID 0
        id      = <10.1.10.1>
        status  = ring 0 active with no faults
RING ID 1
        id      = <10.1.10.2>
        status  = ring 1 active with no faults
```

Prepare for resource creation

To ensure that the cluster does not perform any unexpected actions during setup of resources and configuration, set the maintenance mode to true.

Run the following command to put the cluster in maintenance mode.

```
crm maintenance on
```

Reset configuration – *optional*

> **Note**
>
> The following instructions help you reset the complete configuration. Run these commands only if you want to start setup from the beginning. You can make minor changes with the `crm edit` command.

Run the following command to back up the current configuration for reference.

```
crm config show > /tmp/crmconfig_backup.txt
```

Run the following command to clear the current configuration.

```
crm configure erase
```

⚠ **Important**

Once the preceding erase command is executed, it removes all of the cluster resources from Cluster Information Base (CIB), and disconnects the communication from `corosync` to the cluster. Before starting the resource configuration run `crm cluster restart`, so that cluster reestablishes communication with `corosync`, and retrieves the configuration. The restart of cluster removes *maintenance mode*. Reapply before commencing additional configuration and resource setup.

Cluster bootstrap

Configure the cluster bootstrap parameters by running the following commands.

```
crm configure rsc_defaults resource-stickiness=1
crm configure rsc_defaults migration-threshold=3
crm configure property stonith-enabled="true"
crm configure property stonith-action="off"
crm configure property stonith-timeout="300s"
crm configure op_defaults timeout="300s"
crm configure op_defaults record-pending="true"
```

Add the following command for ENSA2 only. Priority fencing is not possible with ENSA1 constraint priority settings.

```
crm configure property priority-fencing-delay="30s"
```

Create Amazon EC2 STONITH resource

Modify the commands in the following table to match your configuration values.

Example

ENSA1

- Use the following command to create STONITH resource.

  ```
  crm configure primitive res_AWS_STONITH stonith:external/ec2 op start interval=0
  ```

```
timeout=180s op stop interval=0 timeout=180s op monitor interval=180s timeout=60s
 params
tag=<pacemaker> profile=<cluster> pcmk_delay_max=30
```

profile – this refers to the AWS CLI profile crated during setup. In the preceding command, *cluster* is the profile name.

ENSA2

- Use the following command to create STONITH resource with priority fencing delay.

```
crm configure primitive res_AWS_STONITH stonith:external/ec2 op start interval=0
timeout=180s op stop interval=0 timeout=180s op monitor interval=180s timeout=60s
 params
tag=<pacemaker> profile=<cluster> pcmk_delay_max=10
```

profile – this refers to the AWS CLI profile crated during setup. In the preceding command, *cluster* is the profile name.

Create file system resources (classic only)

In classic configuration, the mounting and unmounting of file system resources to align with the location of the SAP services is done using cluster resources.

Modify and run the following commands to create the file system resources.

ASCS

Use the following command to create ASCS file system resources.

```
crm configure primitive rsc_fs_<SLX>_ASCS<00> ocf:heartbeat:Filesystem params
device="<nfs.fqdn>:/<SLX>_ASCS<00>" directory="/usr/sap/<SLX>/ASCS<00>" fstype="nfs4"
options=<"rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2"> op start timeout=60s
interval=0 op stop timeout=60s interval=0 op monitor interval=20s timeout=40s
```

ERS

Use the following command to create ERS file system resources.

```
crm configure primitive rsc_fs_<SLX>_ERS<10> ocf:heartbeat:Filesystem params
device="<nfs.fqdn>:/<SLX>_ERS<10>" directory="/usr/sap/<SLX>/ERS<10>" fstype="nfs4"
```

```
options=<"rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2"> op start timeout=60s
interval=0 op stop timeout=60s interval=0 op monitor interval=20s timeout=40s
```

Notes

- Review the mount options to ensure that they match with your operating system, NFS file system type, and the latest recommendations from SAP.
- <nfs.fqdn> can either be an alias or the default file system resource name of the NFS or FSx for ONTAP resource. For example, `fs-xxxxxx.efs.xxxxxx.amazonaws.com`.

Create overlay IP resources

The IP resource provides the details necessary to update the route table entry for overlay IP.

Modify and run the commands in the table to create IP resources.

ASCS

Use the following command to create an ASCS IP resource.

```
crm configure primitive rsc_ip_<SLX>_ASCS<00> ocf:heartbeat:aws-vpc-move-ip params
ip=<172.16.30.5> routing_table=<rtb-xxxxxroutetable1> interface=eth0 profile=<cluster>
 op start
interval=0 timeout=180s op stop interval=0 timeout=180s op monitor interval=20s
 timeout=40s
```

ERS

Use the following command to create an ERS IP resource.

```
crm configure primitive rsc_ip_<SLX>_ERS<10> ocf:heartbeat:aws-vpc-move-ip params
 ip=<172.16.30.6>
routing_table=<rtb-xxxxxroutetable1> interface=eth0 profile=<cluster> op start
 interval=0
timeout=180s op stop interval=0 timeout=180s op monitor interval=20s timeout=40s
```

Notes

- If more than one route table is required for connectivity or because of subnet associations, the `routing_table` parameter can have multiple values separated by a comma. For example, `routing_table=rtb-xxxxxroutetable1, rtb-xxxxxroutetable2`.

- Additional parameters – `lookup_type` and `routing_table_role` are required for shared VPC. For more information, see {https---docs-aws-amazon-com-sap-latest-sap-netweaver-sles-netweaver-ha-settings-html-sles-netweaver-ha-shared-vpc}[Shared VPC – optional].

Create `sapstartsrv` resources (simple-mount only)

In simple-mount architecture, the `sapstartsrv` process that is used to control start/stop and monitoring of an SAP instance, is controlled by a cluster resource. This new resource adds additional control that removes the requirement for file system resources to be restricted to a single node.

Modify and run the commands in the table to create `sapstartsrv` resource.

ASCS

Use the following command to create an ASCS `sapstartsrv` resource.

```
crm configure primitive rsc_sapstart_<SLX>_ASCS<00> ocf:suse:SAPStartSrv params
InstanceName=<SLX>_ASCS<00>_<slxascs>
```

ERS

Use the following command to create an ERS `sapstartsrv` resource.

```
crm configure primitive rsc_sapstart_<SLX>_ERS<10> ocf:suse:SAPStartSrv params
InstanceName=<SLX>_ERS<10>_<slxers>
```

Create SAP instance resources (simple-mount only)

The minor difference in creating SAP instance resources between classic and simple-mount configurations is the addition of `MINIMAL_PROBE=true` parameters.

The SAP instance is started and stopped using cluster resources. Modify and run the commands in the table to create SAP instance resources.

Example

ENSA1

- Use the following command to create an **ASCS** SAP instance resource.

```
crm configure primitive rsc_sap_<SLX>_ASCS<00> ocf:heartbeat:SAPInstance
 operations
\$id=rsc_sap_<SLX>_ASCS<00>-operations op monitor interval=11s timeout=60s on-
fail=restart op
stop interval=0s timeout=240s params InstanceName=<SLX>_ASCS<00>_<slxascs>
START_PROFILE="/usr/sap/SLX/SYS/profile/<SLX_ASCS00_slxascs>"
 AUTOMATIC_RECOVER=false
MINIMAL_PROBE=true meta resource-stickiness=5000 failure-timeout=60 migration-
threshold=1
priority=10
```

Use the following command to create an **ERS** SAP instance resource.

```
crm configure primitive rsc_sap_<SLX>_ERS<10> ocf:heartbeat:SAPInstance operations
\$id=rsc_sap_<SLX>_ERS<10>-operations op monitor interval=11s timeout=60s on-
fail=restart op
stop interval=0s timeout=240s params InstanceName=<SLX_ERS10_slxers>
START_PROFILE="/usr/sap/<SLX>/SYS/profile/<SLX_ERS10_slxers<"
 AUTOMATIC_RECOVER=false
MINIMAL_PROBE=true IS_ERS=true meta priority=1000
```

ENSA2

- Use the following command to create an **ASCS** SAP instance resource.

```
crm configure primitive rsc_sap_<SLX>_ASCS<00> ocf:heartbeat:SAPInstance
 operations
\$id=rsc_sap_SLX_ASCS00-operations op monitor interval=11s timeout=60s on-
fail=restart op
stop interval=0s timeout=240s params InstanceName=<SLX_ASCS00_slxascs>
START_PROFILE="/usr/sap/<SLX>/SYS/profile/<SLX_ASCS00_slxascs>"
 AUTOMATIC_RECOVER=false
MINIMAL_PROBE=true meta resource-stickiness=5000 priority=1000
```

Use the following command to create an **ERS** SAP instance resource.

```
crm configure primitive rsc_sap_SLX_ERS10 ocf:heartbeat:SAPInstance operations
\$id=rsc_sap_SLX_ERS10-operations op monitor interval=11s timeout=60s on-
fail=restart op
stop interval=0s timeout=240s params InstanceName=<SLX_ERSSLX_slxers>
```

```
START_PROFILE="/usr/sap/<SLX>/SYS/profile/SLX_ERS10_slxers"
 AUTOMATIC_RECOVER=false
MINIMAL_PROBE=true IS_ERS=true
```

The difference between ENSA1 and ENSA2 is that ENSA2 allows the lock table to be consumed remotely, which means that for ENSA2, ASCS can restart in its current location (assuming the node is still available). This change impacts stickiness, migration and priority parameters. Ensure that you use the right command for your enqueue version.

Create SAP instance resources (classic only)

The SAP instance is started and stopped using cluster resources. Modify and run the commands in the table to create SAP instance resources.

Example

ENSA1

- Use the following command to create an **ASCS** SAP instance resource.

```
crm configure primitive rsc_sap_<SLX>_ASCS<00> ocf:heartbeat:SAPInstance
 operations
\$id=rsc_sap_<SLX>_ASCS<00>-operations op monitor interval=11s timeout=60s on-
fail=restart op
stop interval=0s timeout=240s params InstanceName=<SLX_ASCS00_slxascs>
START_PROFILE="/usr/sap/<SLX>/SYS/profile/<SLX_ASCS00_slxascs>"
 AUTOMATIC_RECOVER=false meta
resource-stickiness=5000 failure-timeout=60 migration-threshold=1 priority=10
```

Use the following command to create an **ERS** SAP instance resource.

```
crm configure primitive rsc_sap_<SLX>_ERS<10> ocf:heartbeat:SAPInstance operations
\$id=rsc_sap_<SLX>_ERS<10>-operations op monitor interval=11s on-fail=restart
 timeout=60s op
stop interval=0s timeout=240s on-fail=restart params
 InstanceName=<SLX_ERS10_slxers>
START_PROFILE="/usr/sap/<SLX>/SYS/profile/<SLX_ERS10_slxers>"
 AUTOMATIC_RECOVER=false
IS_ERS=true meta priority=1000
```

ENSA2

- Use the following command to create an **ASCS** SAP instance resource.

```
crm configure primitive rsc_sap_<SLX>_ASCS<00> ocf:heartbeat:SAPInstance
 operations
\$id=rsc_sap_<SLX>_ASCS<00>-operations op monitor interval=11s timeout=60s on-
fail=restart op
stop interval=0s timeout=240s params InstanceName=<SLX_ASCS00_slxascs>
START_PROFILE="/usr/sap/<SLX>/SYS/profile/<SLX_ASCS00_slxascs>"
 AUTOMATIC_RECOVER=false meta
resource-stickiness=5000 priority=1000
```

Use the following command to create an **ERS** SAP instance resource.

```
crm configure primitive rsc_sap_<SLX>_ERS<10> ocf:heartbeat:SAPInstance operations
\$id=rsc_sap_<SLX>_ERS<10>-operations op monitor interval=11s timeout=60s on-
fail=restart op
stop interval=0s timeout=240s params InstanceName=<SLX_ERS10_slxers>
START_PROFILE="/usr/sap/<SLX>/SYS/profile/<SLX_ERS10_slxers>"
 AUTOMATIC_RECOVER=false
IS_ERS=true
```

The change between ENSA1 and ENSA2 allows the lock table to be consumed remotely. If the node is still available, ASCS can restart in its current location for ENSA2. This impacts stickiness, migration, and priority parameters. Make sure to use the right command, depending on your enqueue server.

Create cluster resource groups for `ip / sapstart / sap` (simple-mount only)

A cluster resource group is a set of resources that need to be located together, start sequentially, and stopped in the reverse order.

In simple-mount architecture, the overlay IP must be available first, then the SAP services are started before the SAP instance can start. Modify and run the commands in the table to create cluster resource groups.

ASCS

Use the following command to create an ASCS cluster resource group.

```
crm configure group grp_<SLX>_ASCS<00> rsc_ip_<SLX>_ASCS
rsc_sapstart_SLX_ASCS00 rsc_sap_<SLX>_ASCS<00> meta resource-stickiness=3000
```

ERS

Use the following command to create an ERS cluster resource group.

```
crm configure group grp_<SLX>_ERS<10> rsc_ip_<SLX>_ERS<10> rsc_sapstart_<SLX>_ERS<10>
 rsc_sap_<SLX>_ERS
```

Create cluster resource groups for `ip/sapstart/sap` (classic only)

A cluster resource group is a set of resources that need to be located together, start sequentially, and stopped in the reverse order.

In classic architecture, the file system is mounted first, then the overlay IP must be available before the SAP instance can start. Modify and run the commands in the table to create cluster resource groups.

ASCS

Use the following command to create an ASCS cluster resource group.

```
crm configure group grp_<SLX>_ASCS<00> rsc_fs_<SLX>_ASCS<00> rsc_ip_<SLX>_ASCS<00>
 rsc_sap_<SLX>_ASCS
meta resource-stickiness=3000
```

ERS

Use the following command to create an ERS cluster resource group.

```
crm configure group grp_<SLX>_ERS<10> rsc_fs_<SLX>_ERS<10> rsc_ip_<SLX>_ERS<10>
 rsc_sap_<SLX>_ERS
```

Create resource constraints

Resource constraints are used to determine where resources run per the conditions. Constraints for SAP NetWeaver ensure that ASCS and ERS are started on separate nodes and locks are preserved in case of failures. The following are the different types of constraints.

Colocation constraint

The negative score ensures that ASCS and ERS are run on separate nodes, wherever possible.

```
crm configure colocation col_sap_<SLX>_separate -5000: grp_<SLX>_ERS<10>
 grp_<SLX>_ASCS
```

Order constraint

This constraint ensures the ASCS instance is started prior to stopping the ERS instance. This is necessary to consume the lock table.

```
crm configure order ord_sap_<SLX>_first_start_ascs Optional:
 rsc_sap_<SLX>_ASCS<00>:start rsc_sap_<SLX>_ERS<10>:stop symmetrical=false
```

Location constraint (ENSA1 only)

This constraint is only required for ENSA1. The lock table can be retrieved remotely for ENSA2, and as a result ASCS doesn't failover to where ERS is running.

```
crm configure location loc_sap_<SLX>_failover_to_ers rsc_sap_<SLX>_ASCS<00> rule 2000:
 runs_ers_<SLX> eq 1
```

Activate cluster

Use `crm config show` and `crm config edit` commands to review that all the values have been entered correctly.

On confirmation of correct values, set the maintenance mode to false using the following command. This enables the cluster to take control of the resources.

```
crm maintenance off
```

See the Sample configuration.

Sample configuration (`crm config`)

The following sample configuration is based on simple mount and ENSA2.

```
node 1: slxhost01 \
        attributes standby=off
node 2: slxhost02 \
        attributes standby=off
primitive res_{aws}_STONITH stonith:external/ec2 \
        op start interval=0 timeout=180s \
        op stop interval=0 timeout=180s \
        op monitor interval=180s timeout=60s \
        params tag=pacemaker profile=cluster pcmk_delay_max=10
primitive rsc_ip_SLX_ASCS00 aws-vpc-move-ip \
        params ip=172.16.52.5 routing_table=rtb-04385ff6bcae17982 interface=eth0
 profile=cluster \
        op start interval=0 timeout=180s \
        op stop interval=0 timeout=180s \
        op monitor interval=20s timeout=40s
primitive rsc_ip_SLX_ERS10 aws-vpc-move-ip \
        params ip=172.16.52.6 routing_table=rtb-04385ff6bcae17982 interface=eth0
 profile=cluster \
        op start interval=0 timeout=180s \
        op stop interval=0 timeout=180s \
        op monitor interval=20s timeout=40s
primitive rsc_sap_SLX_ASCS00 SAPInstance \
        operations $id=rsc_sap_SLX_ASCS00-operations \
        op monitor interval=11s timeout=60s on-fail=restart \
        op stop interval=0 timeout=240s \
        params InstanceName=SLX_ASCS00_slxascs START_PROFILE=" /usr/sap/SLX/SYS/
profile/SLX_ASCS00_slxascs" AUTOMATIC_RECOVER=false MINIMAL_PROBE=true \
        meta resource-stickiness=5000 priority=1000
primitive rsc_sap_SLX_ERS10 SAPInstance \
        operations $id=rsc_sap_SLX_ERS10-operations \
        op monitor interval=11s timeout=60s on-fail=restart \
        op stop interval=0 timeout=240s \
        params InstanceName=SLX_ERS10_slxers START_PROFILE=" /usr/sap/SLX/SYS/profile/
SLX_ERS10_slxers" AUTOMATIC_RECOVER=false IS_ERS=true MINIMAL_PROBE=true
primitive rsc_sapstart_SLX_ASCS00 ocf:suse:SAPStartSrv \
        params InstanceName=SLX_ASCS00_slxascs
primitive rsc_sapstart_SLX_ERS10 ocf:suse:SAPStartSrv \
        params InstanceName=SLX_ERS10_slxers
group grp_SLX_ASCS00 rsc_ip_SLX_ASCS00 rsc_sapstart_SLX_ASCS00 rsc_sap_SLX_ASCS00 \
        meta resource-stickiness=3000
group grp_SLX_ERS10 rsc_ip_SLX_ERS10 rsc_sapstart_SLX_ERS10 rsc_sap_SLX_ERS10
colocation col_sap_SLX_separate -5000: grp_SLX_ERS10 grp_SLX_ASCS00
order ord_sap_SLX_first_start_ascs Optional: rsc_sap_SLX_ASCS00:start
 rsc_sap_SLX_ERS10:stop symmetrical=false
property cib-bootstrap-options: \
        maintenance-mode=false \
        stonith-enabled=true \
        stonith-action=off \
        stonith-timeout=300s \
        priority-fencing-delay=30s \
        have-watchdog=false \
        dc-version="2.0.5+20201202.ba59be712-150300.4.24.1-2.0.5+20201202.ba59be712" \
        cluster-infrastructure=corosync \
        last-lrm-refresh=1670459760
rsc_defaults rsc-options: \
        resource-stickiness=1 \
        migration-threshold=3
op_defaults op-options: \
        timeout=600s \
        record-pending=true
```

Operations

This section covers the following topics.

Topics

- Analysis and maintenance
- Testing

Analysis and maintenance

This section covers the following topics.

Topics

- Viewing the cluster state
- Performing planned maintenance
- Post-failure analysis and reset
- Alerting and monitoring

Viewing the cluster state

You can view the state of the cluster in two ways - based on your operating system or with a web based console provided by SUSE.

Topics

- Operating system based
- SUSE Hawk2

Operating system based

There are multiple operating system commands that can be run as root or as a user with appropriate permissions. The commands enable you to get an overview of the status of the cluster and its services. See the following commands for more details.

```
crm status
```

Sample output:

```
slxhost01:~ # crm status
Cluster Summary:
  * Stack: corosync
  * Current DC: slxhost01 (version
 2.0.5+20201202.ba59be712-150300.4.24.1-2.0.5+20201202.ba59be712) - partition with
 quorum
  * Last updated: Tue Nov  1 13:41:58 2022
  * Last change:  Fri Oct 28 08:55:43 2022 by root via crm_attribute on slxhost02
  * 2 nodes configured
  * 7 resource instances configured

Node List:
  * Online: [ slxhost01 slxhost02 ]

Full List of Resources:
  * Resource Group: grp_SLX_ASCS00:
    * rsc_ip_SLX_ASCS00 (ocf::heartbeat:aws-vpc-move-ip):        Started slxhost01
    * rsc_sapstart_SLX_ASCS00   (ocf::suse:SAPStartSrv):         Started slxhost01
    * rsc_sap_SLX_ASCS00        (ocf::heartbeat:SAPInstance):    Started slxhost01
  * res_AWS_STONITH     (stonith:external/ec2):  Started slxhost02
  * Resource Group: grp_SLX_ERS10:
    * rsc_ip_SLX_ERS10  (ocf::heartbeat:aws-vpc-move-ip):        Started slxhost02
    * rsc_sapstart_SLX_ERS10    (ocf::suse:SAPStartSrv):         Started slxhost02
    * rsc_sap_SLX_ERS10 (ocf::heartbeat:SAPInstance):    Started slxhost02
```

The following table provides a list of useful commands.

Command	Description
`crm_mon`	Display cluster status on the console with updates as they occur
`crm_mon -1`	Display cluster status on the console just once, and exit
`crm_mon -Arnf`	-A Display node attributes -n Group resources by node -r Display inactive resources -f Display resource fail counts
`crm help`	View more options
`crm_mon --help-all`	View more options

SUSE Hawk2

Hawk2 is a web-based graphical user interface for managing and monitoring pacemaker highly availability clusters. It must be enabled on every node in the cluster, to point your web browser on any node for accessing it. Use the following command to enable Hawk2.

```
systemctl enable --now hawk
systemctl status hawk
```

Use the following URL to check security groups for access on port 7630 from your administrative host.

```
https://your-server:7630/

e.g https://slxhost01:7630
```

For more information, see Configuring and Managing Cluster Resources with Hawk2 in the SUSE Documentation.

Performing planned maintenance

The cluster connector is designed to integrate the cluster with SAP start framework (`sapstartsrv`), including the rolling kernel switch (RKS) awareness. Stopping and starting the SAP system using `sapcontrol` should not result in any cluster remediation activities as these actions are not interpreted as failures. Validate this scenario when testing your cluster.

There are different options to perform planned maintenance on nodes, resources, and the cluster.

Topics

- Maintenance mode
- Placing a node in standby mode
- Moving a resource (not recommended)

Maintenance mode

Use maintenance mode if you want to make any changes to the configuration or take control of the resources and nodes in the cluster. In most cases, this is the safest option for administrative tasks.

Example

On

Use one of the following commands to turn on maintenance mode.

```
crm maintenance on
```

```
crm configure property maintenance-mode="true"
```

Off

Use one of the following commands to turn off maintenance mode.

```
crm maintenance off
```

```
crm configure property maintenance-mode="false"
```

Placing a node in standby mode

To perform maintenance on the cluster without system outage, the recommended method for moving active resources is to place the node you want to remove from the cluster in standby mode.

```
crm node standby <slxhost01>
```

The cluster will cleanly relocate resources, and you can perform activities, including reboots on the node in standby mode. When maintenance activities are complete, you can re-introduce the node with the following command.

```
crm node online <slxhost01>
```

Moving a resource (not recommended)

Moving individual resources is not recommended because of the migration or move constraints that are created to lock the resource in its new location. These can be cleared as described in the info messages, but this introduces an additional setup.

```
<slxhost01>:~ crm resource move grp_<SLX>_ASCS<00> <slxhost02>
INFO: Move constraint created for grp_<SLX>_ASCS<00> to <slxhost02>
INFO: Use `crm resource clear grp_<SLX>_ASCS<00>` to remove this constraint
```

Use the following command once the resources have relocated to their target location.

```
<slxhost01>:~ crm resource clear grp_SLX_ASCS00
```

Post-failure analysis and reset

A review must be conducted after each failure to understand the source of failure as well the reaction of the cluster. In most scenarios, the cluster prevents an application outage. However, a manual action is often required to reset the cluster to a protective state for any subsequent failures.

Topics

- Checking the logs
- Cleanup crm status
- Restart failed nodes or pacemaker

- Further analysis

Checking the logs

Start your troubleshooting by checking the operating system log /var/log/messages. You can find additional information in the cluster and pacemaker logs.

- **Cluster logs** – updated in the corosync.conf file located at /etc/corosync/corosync.conf.
- **Pacemaker logs** – updated in the pacemaker.log file located at /var/log/pacemaker.
- **Resource agents** – /var/log/messages

Application based failures can be investigated in the SAP work directory.

Cleanup crm status

If failed actions are reported using the crm status command, and if they have already been investigated, then you can clear the reports with the following command.

```
crm resource cleanup <resource> <hostname>
```

Restart failed nodes or pacemaker

It is recommended that failed (or fenced) nodes are not automatically restarted. It gives operators a chance to investigate the failure, and ensure that the cluster doesn't make assumptions about the state of resources.

You need to restart the instance or the pacemaker service based on your approach.

Further analysis

The following commands consolidate information from both nodes, highlighting key events and differentiating between originating node to make the analysis clear.

```
crm history events

crm history log
```

If further analysis from SUSE is required, an hb_report may be requested. For more information, see SUSE Documentation – Usage of hb_report for SLES HAE.

Note

`crm history events` and `hb_report` rely on passwordless ssh being set up between the nodes.

Alerting and monitoring

This section covers the following topics.

Topics

- Using Amazon CloudWatch Application Insights
- Using the cluster alert agents

Using Amazon CloudWatch Application Insights

For monitoring and visibility of cluster state and actions, Application Insights includes metrics for monitoring enqueue replication state, cluster metrics, and SAP and high availability checks. Additional metrics, such as EFS and CPU monitoring can also help with root cause analysis.

For more information, see Get started with Amazon CloudWatch Application Insights and SAP NetWeaver High Availability on Amazon EC2.

Using the cluster alert agents

Within the cluster configuration, you can call an external program (an alert agent) to handle alerts. This is a *push* notification. It passes information about the event via environment variables.

The agents can then be configured to send emails, log to a file, update a monitoring system, etc. For example, the following script can be used to access Amazon SNS.

```
#!/bin/sh

alert_sns.sh
modified from /usr/share/pacemaker/alerts/alert_smtp.sh.sample

##############################################################################
#SETUP
* Create an SNS Topic and subscribe email or chatbot
* Note down the ARN for the SNS topic
* Give the IAM Role attached to both Instances permission to publish to the SNS Topic
* Ensure the aws cli is installed
* Copy this file to /usr/share/pacemaker/alerts/alert_sns.sh or other location on BOTH
 nodes
* Ensure the permissions allow for hacluster and root to execute the script
* Run the following as root (modify file location if necessary and replace SNS ARN):

#SLES:
crm configure alert aws_sns_alert /usr/share/pacemaker/alerts/alert_sns.sh meta
 timeout=30s timestamp-format="%Y-%m-%d_%H:%M:%S" to <{ arn:aws:sns:region:account-
id:myPacemakerAlerts  }>
#RHEL:
pcs alert create id=aws_sns_alert path=/usr/share/pacemaker/alerts/alert_sns.sh meta
 timeout=30s timestamp-format="%Y-%m-%d_%H:%M:%S"
pcs alert recipient add aws_sns_alert value=arn:aws:sns:region:account-
id:myPacemakerAlerts

#Additional information to send with the alerts.
node_name=`uname -n`
sns_body=`env | grep CRM_alert_`

#Required for SNS
TOKEN=$(/usr/bin/curl --noproxy '*' -s -X PUT "http://169.254.169.254/latest/api/token"
 -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

#Get metadata
REGION=$(/usr/bin/curl --noproxy '*' -w "\n" -s -H "X-aws-ec2-metadata-token: $TOKEN"
 http://169.254.169.254/latest/dynamic/instance-identity/document | grep region | awk -
F\" '{print $4}')

sns_subscription_arn=${CRM_alert_recipient}

#Format depending on alert type
case ${CRM_alert_kind} in
   node)
     sns_subject="${CRM_alert_timestamp} ${cluster_name}: Node '${CRM_alert_node}' is
 now '${CRM_alert_desc}'"
    ;;
   fencing)
     sns_subject="${CRM_alert_timestamp} ${cluster_name}: Fencing ${CRM_alert_desc}"
    ;;
   resource)
     if [ ${CRM_alert_interval} = "0" ]; then
         CRM_alert_interval=""
     else
```

```
         CRM_alert_interval=" (${CRM_alert_interval})"
     fi
     if [ ${CRM_alert_target_rc} = "0" ]; then
         CRM_alert_target_rc=""
     else
         CRM_alert_target_rc=" (target: ${CRM_alert_target_rc})"
     fi
     case ${CRM_alert_desc} in
         Cancelled)
           ;;
         *)
           sns_subject="${CRM_alert_timestamp}: Resource operation
 '${CRM_alert_task}${CRM_alert_interval}' for '${CRM_alert_rsc}' on
 '${CRM_alert_node}': ${CRM_alert_desc}${CRM_alert_target_rc}"
           ;;
     esac
     ;;
   attribute)
     sns_subject="${CRM_alert_timestamp}: The '${CRM_alert_attribute_name}' attribute
 of the '${CRM_alert_node}' node was updated in '${CRM_alert_attribute_value}'"
     ;;
   *)
     sns_subject="${CRM_alert_timestamp}: Unhandled $CRM_alert_kind alert"
     ;;
esac

#Use this information to send the email.
aws sns publish --topic-arn "${sns_subscription_arn}" --subject "${sns_subject}" --
message "${sns_body}" --region ${REGION}
```

Testing

We recommend scheduling regular fault scenario recovery testing at least annually, and as part of the operating system or SAP kernel updates that may impact operations. For more details on best practices for regular testing, see SAP Lens – Best Practice 4.3 – Regularly test business continuity plans and fault recovery.

The tests described here simulate failures. These can help you understand the behavior and operational requirements of your cluster.

In addition to checking the state of cluster resources, ensure that the service you are trying to protect is in the required state. Can you still connect to SAP? Are locks still available in SM12?

Define the recovery time to ensure that it aligns with your business objectives. Record recovery actions in runbooks.

Topics

Test 1: Stop ASCS on the primary node using `sapcontrol`

Notes – Ensure that the connector has been installed and the parameters have been updated.

Simulate failure – On `slxhost01` as `slxadm`:

```
sapcontrol -nr <00> -function Stop
```

Expected behavior – ASCS should be stopped on `slxhost01`, and the cluster should not perform any activity.

Recovery action – Start ASCS manually.

Test 2: Stop ERS on the secondary node using `sapcontrol`

Notes – Ensure that the connector has been installed, and the parameters are updated.

Simulate failure – On `slxhost02` as `slxadm`:

```
sapcontrol -nr <10> -function Stop
```

Expected behavior – ERS should be stopped on `slxhost02`, and the cluster should not perform any activity.

Recovery action – Start ERS manually.

Test 3: Kill the message server process on the primary node

Simulate failure – On `slxhost01` as `slxadm`:

```
kill -9 $(pgrep -f "ms.sap<SLX>_ASCS<00>")
```

Expected behavior – The message server should immediately respawn based on the Restart parameter.

Recovery action – No action required.

Test 4: Kill the enqueue server process on the primary node

Notes – Check that locks have persisted, and review the location constraints that only exist for ENSA1.

Simulate failure – On `slxhost01` as `slxadm`:

```
kill -9 $(pgrep -f "[en|enq].sap<SLX>_ASCS<00>")
```

Expected behavior – ENSA2: Cluster will restart the ENQ process and retrieve the locks remotely. ENSA1: Cluster will failover the ASCS resource to the node where the ERS is running.

Recovery action – No action required.

Test 5: Kill the ER process

Simulate failure – On `slxhost02` as `slxadm`:

```
kill -9 $(pgrep -f "[er|enqr].sap<SLX>_ERS<10>")
```

Expected behavior – Cluster will restart the ERS on the same node.

Recovery action – No action required.

Test 6: Simulate hardware failure of an individual node, and repeat for other node

Notes – To simulate a system crash, you must first ensure that `/proc/sys/kernel/sysrq` is set to 1.

Simulate failure – On both nodes as root:

```
echo 'b' > /proc/sysrq-trigger
```

Expected behavior – The node which has been killed fails. The cluster will move the resources (ASCS/ERS) which were running on the failed node to the surviving node.

Recovery action – Start the EC2 node and pacemaker service. The cluster will detect that the node is online and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Test 7: Simulate a network failure

Notes – See the following list.

- Iptables must be installed.
- Use a subnet in this command because of the secondary ring.
- Check for any existing iptables rules as iptables -F will flush all rules.
- Review pcmk_delay and priority parameters if you see neither node survives the fence race.

Simulate failure – On either node as root:

```
iptables -A INPUT -s <CIDR_of_other_subnet> -j DROP; iptables -A OUTPUT -d
 <CIDR_of_other_subnet> -j DROP
```

Expected behavior – The cluster detects the network failure, and fences one of the nodes to avoid a split-brain situation.

Recovery action – If the node where the command was run survives, execute iptables -F to clear the network failure. Start the EC2 node and pacemaker service. The cluster will detect that the node is online and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Test 8: Simulate an NFS failure

Notes – See the following list.

- Iptables must be installed.

- Check for any existing iptables rules as iptables -F will flush all rules.
- Although rare, this is an important scenario to test. Depending on the activity it may take some time (10 min +) to notice that I/O to EFS is not occurring and fail either the Filesystem or SAP resources.

Simulate failure – On either node as root:

```
iptables -A OUTPUT -p tcp --dport 2049 -m state --state NEW,ESTABLISHED,RELATED -j
 DROP; iptables -A INPUT -p tcp --sport 2049 -m state --state ESTABLISHED -j DROP
```

Expected behavior – The cluster detects that NFS is not available, and the SAP Instance resource agent will fail and move to the FAILED state. Because of the option "on-fail=restart" configuration, the cluster will try a local restart before eventually fencing the node and failing over.

Recovery action – If the node where the command was run survives, execute iptables -F to clear the network failure. Start the EC2 node and pacemaker service. The cluster will detect that the node is online and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Test 9: Accidental shutdown

Notes – See the following list.

- Avoid shutdowns without cluster awareness.
- We recommend the use of systemd to ensure predictable behaviour.
- Ensure the resource dependencies are in place.

Simulate failure – Login to AWS Management Console, and stop the instance or issue a shutdown command.

Expected behavior – The node which has been shut down fails. The cluster will move the resources (ASCS/ERS) which were running on the failed node to the surviving node. If systemd and resource dependencies are not configured, you may notice that while the EC2 instance is shutting down gracefully, the cluster will detect an unclean stop of cluster services on the node and will fence the EC2 instance being shut down. For more information, see SUSE documentation – Stopping the Cluster Services on a Node.

Recovery action – Start the EC2 node and pacemaker service. The cluster will detect that the node is online, and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

SAP NetWeaver on AWS: high availability configuration for Red Hat Enterprise Linux (RHEL) for SAP applications

This topic applies to Red Hat Enterprise Linux (RHEL) for SAP with high availability and update services operating system for SAP NetWeaver applications on AWS cloud. It covers the instructions for configuration of a pacemaker cluster for the ABAP SAP Central Service (ASCS) and the Enqueue Replication Server (ERS) when deployed on Amazon EC2 instances in two different Availability Zones within an AWS Region.

This guide includes the following options.

- Standalone Enqueue Server 1 (ENSA1) and Standalone Enqueue Server 2 (ENSA2)
- RHEL 7.x and RHEL 8.x, where commands vary

Topics

Planning

This section covers the following topics.

Topics

Prerequisites

You must meet the following prerequisites before commencing setup.

Topics

- Deployed cluster infrastructure
- Supported operating system
- Required access for setup

Deployed cluster infrastructure

Ensure that your AWS networking requirements and Amazon EC2 instances where SAP workloads are installed, are correctly configured for SAP. For more information, see SAP NetWeaver Environment Setup for Linux on AWS.

See the following ASCS cluster specific requirements.

- Two cluster nodes created in private subnets in separate Availability Zones within the same Amazon VPC and AWS Region
- Access to the route table(s) that are associated with the chosen subnets

  For more information, see AWS – Overlay IP.
- Targeted Amazon EC2 instances must have connectivity to the Amazon EC2 endpoint via internet or a Amazon VPC endpoint.

Supported operating system

Protecting ASCS with a pacemaker cluster requires packages from Red Hat, including targeted cluster resource agents for SAP and AWS that may not be available in standard repositories.

SAP and Red Hat recommend the use of Red Hat Enterprise Linux for SAP. Starting with Red Hat Enterprise Linux 8 (RHEL 8), either RHEL for SAP Solutions or RHEL for SAP Applications are required for running SAP applications in production environments. See SAP Note 1656099 - SAP Applications on AWS: Supported DB/OS and Amazon EC2 products (requires SAP portal access).

Built on the Red Hat Enterprise Linux operating system, Red Hat Enterprise Linux for SAP expands existing capabilities, lso you can get the most out of SAP's powerful analytics and data management portfolio. See Red Hat Enterprise Solutions for SAP product page from Red Hat.

Red Hat Enterprise Linux High Availability (HA) provides all the necessary packages for configuring pacemaker-based clusters. Extended Update Support (E4S) provides support on specific minor releases for 4 years from general availability.

Red Hat Enterprise Linux for SAP with HA and US is available on AWS Marketplace under an hourly or an annual subscription model or can be accessed using a BYOS subscription model.

Required access for setup

The following access is required for setting up the cluster.

- An IAM user with the following privileges.
  - modify Amazon VPC route tables
  - modify Amazon EC2 instance properties
  - create IAM policies and roles
  - create Amazon EFS file systems
- Root access to the operating system of both cluster nodes
- SAP administrative user access – `<sid>adm`

  In case of a new install, this user is created by the install process.

Reliability

ASCS is a single point of failure in a highly available SAP architecture. We recommend evaluating the impact of design decisions on cost, operation, and reliability. For more information, see Reliability in SAP Lens - AWS Well-Architected Framework.

SAP and Red Hat references

In addition to this guide, see the following references for more details.

- Red Hat documentation – SAP S/4 HANA - Enqueue Replication 2 High Availability Cluster - Setup Guide
- Red Hat documentation – SAP NetWeaver - Enqueue Replication 1 High Availability Cluster - Setup Guide
- Red Hat documentation – Red Hat Enterprise Linux for SAP offerings on Amazon Web Services FAQ

- SAP Note: 1656099 - SAP Applications on AWS: Supsported DB/OS and Amazon EC2 products
- SAP Note 1618572 - Linux: Support Statement for RHEL on Amazon Web Services
- SAP Note 2002167 - Red Hat Enterprise Linux 7.x: Installation and Upgrade
- SAP Note 2772999 - Red Hat Enterprise Linux 8.x: Installation and Configuration

You must have SAP portal access for reading all SAP Notes.

Concepts

This section covers AWS, SAP, and Red Hat concepts.

Topics

SAP – ABAP SAP Central Services (ASCS)

The ABAP SAP Central Services (ASCS) is an SAP instance consisting of the following two services. It is considered a single point of failure (SPOF) in a resilient SAP system.

- **Message server** – Responsible for application load distribution (GUI and RFC), communication between application servers, and centralised configuration information for web dispatchers and application servers.
- **Enqueue server (standalone)** – Maintains the lock table in the main memory of ASCS, and is used to synchronize database access. The lock mechanism prevents two transactions from changing the same data in the database simultaneously.

> **Note**
>
> With ABAP Release 7.53 (ABAP Platform 1809), the new Standalone Enqueue Server 2 (ENSA2) is installed by default. It replaces the previous version (ENSA1) but can be

configured for the previous versions. See SAP Note 2630416 - Support for Standalone Enqueue Server 2 (SAP portal access required) for more information.
This document includes modifications to align with the correct ENSA version.

SAP – Enqueue Replication Server (ERS)

The Enqueue Replication Server (ERS) is an SAP instance containing a replica of the lock table (replication table).

In a resilient setup, if the standalone enqueue server (EN/ENQ) fails, it can be restarted either by restart parameters or by high availability software, such as the Pacemaker. The enqueue server retrieves the replication table remotely or by failing over to the host where the ERS is running.

AWS – Availability Zones

Availability Zone is one or more data centers with redundant power, networking, and connectivity in an AWS Region. For more information, see Regions and Availability Zones.

For mission critical deployments of SAP on AWS where the goal is to minimise the recovery time objective (RTO), we suggest distributing single points of failure across Availability Zones. Compared with single instance or single Availability Zone deployments, this increases resilience and isolation against a broad range of failure scenarios and issues, including natural disasters.

Each Availability Zone is physically separated by a meaningful distance (many kilometers) from another Availability Zone. All Availability Zones in an AWS Region re interconnected with high-bandwidth, low-latency network, over fully redundant, dedicated metro fiber. This enables synchronous replication. All traffic between Availability Zones is encrypted.

AWS – Overlay IP

Overlay IP enables a connection to the application, regardless of which Availability Zone (and subnet) contains the active primary node.

When deploying instances in AWS, it is necessary to allocate an IP from a pre-existing subnet. Subnets have a classless inter-domain routing (CIDR) IP assignment from the VPC which resides entirely within one Availability Zone. This CIDR IP assignment cannot span multiple Availability Zones or be reassigned to an instance in a different Availability Zone after faults, including network connectivity or hardware issues which require a failover to the replication target.

To address this, we suggest that you configure an overlay IP, and use this in the connection parameters for the application. This IP address is a non-overlapping RFC1918 private IP address from outside of VPC CIDR block and is configured as an entry in the route table or tables. The route directs the connection to the active node and is updated during a failover by the cluster software.

You can select any one of the following RFC1918 private IP addresses for your overlay IP address.

- 10.0.0.0 – 10.255.255.255 (10/8 prefix)
- 172.16.0.0 – 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 – 192.168.255.255 (192.168/16 prefix)

If you use the 10/8 prefix in your SAP VPC, selecting a 172 or a 192 IP address may help to differentiate the overlay IP. Consider the use of an IP Address Management (IPAM) tool such as Amazon VPC IP Address Manager to plan, track, and monitor IP addresses for your AWS workloads. For more information, see What is IPAM?

The overlay IP agent in the cluster can also be configured to update multiple route tables which contain the Overlay IP entry if your subnet association or connectivity requires it.

Access to overlay IP

The overlay IP is outside of the range of the VPC, and therefore cannot be reached from locations that are not associated with the route table, including on-premises and other VPCs.

Use AWS Transit Gateway as a central hub to facilitate the network connection to an overlay IP address from multiple locations, including Amazon VPCs, other AWS Regions, and on-premises using AWS Direct Connect or AWS Client VPN.

If you do not have AWS Transit Gateway set up as a network transit hub or if it is not available in your preferred AWS Region, you can use a Network Load Balancer to enable network access to an overlay IP.

For more information, see SAP on AWS High Availability with Overlay IP Address Routing.

AWS – Shared VPC

An enterprise landing zone setup or security requirements may require the use of a separate cluster account to restrict the route table access required for the Overlay IP to an isolated account. For more information, see Share your VPC with other accounts.

Evaluate the operational impact against your security posture before setting up shared VPC. To set up, see Shared VPC – optional.

Pacemaker - STONITH fencing agent

In a two-node cluster setup for a primary resource and its replication pair, it is important that there is only one node in the primary role with the ability to modify your data. In the event of a failure scenario where a node is unresponsive or incommunicable, ensuring data consistency that can require you to isolate the faulty node by powering it down before the cluster commences other actions, such as promoting a new primary. This arbitration is the role of the fencing agent.

Since a two-node cluster introduces the possibility of a fence race in which a dual shoot out can occur with communication failures resulting in both nodes simultaneously claiming, "I can’t see you, so I am going to power you off". The fencing agent is designed to minimise this risk by providing an external witness.

Red Hat supports several fencing agents, including the one recommended for use with Amazon EC2 Instances (`fence_aws`). This resource uses API commands to check its own instance status - "Is my instance state anything other than running?" before proceeding to power off its pair. If it is already in a stopping or stopped state it will admit defeat and leave the surviving node untouched.

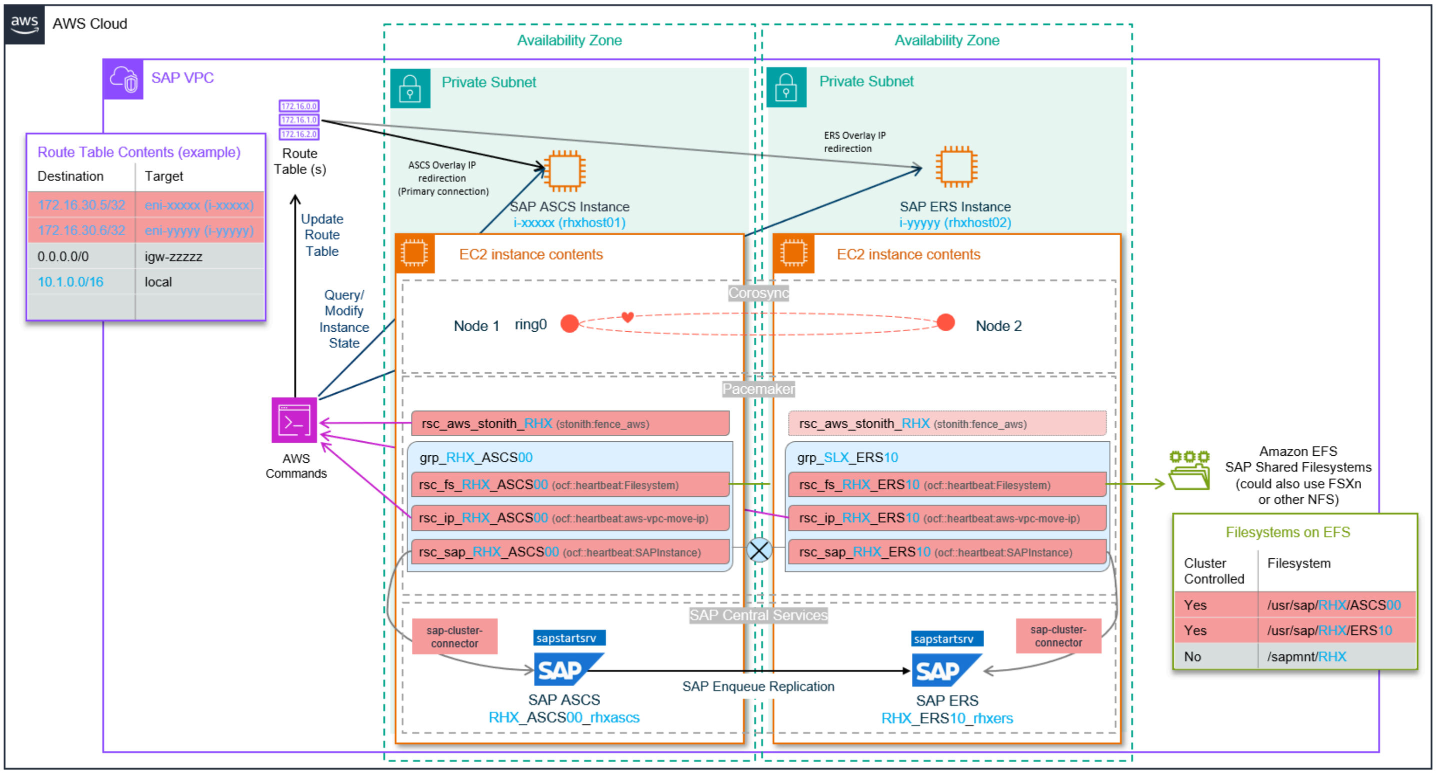
Automation and AWS Launch Wizard for SAP

You can set up a cluster manually using the instructions provided here. You can also automate parts of this process to ensure consistency and repeatability.

Use AWS Launch Wizard for SAP for automated deployments of SAP NetWeaver, SAP S/4 HANA, SAP B/4HANA, and Solution Manager. Launch Wizard uses AWS CloudFormation scripts to quickly provision the resources needed to deploy SAP NetWeaver and S/4 HANA. The automation performs SAP enqueue replication and pacemaker setup so that only validation and testing are required. For more information, see AWS Launch Wizard for SAP.

To ensure that the behavior and operation of your cluster is well understood regardless of how your system is set up, we recommend a thorough test cycle. See Testing for more details.

Architecture diagram


AWS Cloud
Availability Zone
Availability Zone
SAP VPC
Private Subnet
Private Subnet
172.16.0.0
172.16.1.0
172.16.2.0
Route Table (s)
Route Table Contents (example)
Destination | Target
172.16.30.5/32 | eni-xxxxx (i-xxxxx)
172.16.30.6/32 | eni-yyyyy (i-yyyyy)
0.0.0.0/0 | igw-zzzzz
10.1.0.0/16 | local
Update Route Table
Query/ Modify Instance State
AWS Commands
ASCS Overlay IP redirection (Primary connection)
SAP ASCS Instance
i-xxxxx (rhxhost01)
ERS Overlay IP redirection
SAP ERS Instance
i-yyyyy (rhxhost02)
EC2 instance contents
EC2 instance contents
Corosync
Node 1
ring0
Node 2
Pacemaker
rsc_aws_stonith_RHX (stonith:fence_aws)
grp_RHX_ASCS00
rsc_fs_RHX_ASCS00 (ocf::heartbeat:Filesystem)
rsc_ip_RHX_ASCS00 (ocf::heartbeat:aws-vpc-move-ip)
rsc_sap_RHX_ASCS00 (ocf::heartbeat:SAPInstance)
rsc_aws_stonith_RHX (stonith:fence_aws)
grp_SLX_ERS10
rsc_fs_RHX_ERS10 (ocf::heartbeat:Filesystem)
rsc_ip_RHX_ERS10 (ocf::heartbeat:aws-vpc-move-ip)
rsc_sap_RHX_ERS10 (ocf::heartbeat:SAPInstance)
SAP Central Services
sap-cluster-connector
sapstartsrv
SAP
SAP ASCS
RHX_ASCS00_rhxascs
SAP Enqueue Replication
sapstartsrv
sap-cluster-connector
SAP
SAP ERS
RHX_ERS10_rhxers
Amazon EFS
SAP Shared Filesystems
(could also use FSXn or other NFS)
Filesystems on EFS
Cluster Controlled | Filesystem
Yes | /usr/sap/RHX/ASCS00
Yes | /usr/sap/RHX/ERS10
No | /sapmnt/RHX

Deployment

This section covers the following topics.

Topics

- Settings and prerequisites
- SAP and cluster setup
- Cluster configuration

Settings and prerequisites

The cluster setup uses parameters, including `SID` and `System Number` that are unique to your setup. It is useful to predetermine the values with the following examples and guidance.

Topics

- Define reference parameters for setup
- Amazon EC2 instance settings

- Operating system prerequisites
- IP and hostname resolution prerequisites
- File system prerequisites
- Shared VPC – optional

Define reference parameters for setup

The cluster setup relies on the following parameters.

Topics

Global AWS parameters

Name	Parameter	Example
AWS account ID	`<account_id>`	`123456789100`
AWS Region	`<region_id>`	`us-east-1`

- AWS account – For more details, see {https---docs-aws-amazon-com-IAM-latest-UserGuide-console-account-alias-html}[Your AWS account ID and its alias].
- AWS Region – For more details, see {https---docs-aws-amazon-com-AWSEC2-latest-UserGuide-using-regions-availability-zones-html-using-regions-availability-zones-describe}[Describe your Regions].

Amazon EC2 instance parameters

Name	Parameter	Primary example	Secondary example
Amazon EC2 instance ID	`<instance_id>`	`i-xxxxinstidforhost1`	`i- xxxxinstidforhost2`
Hostname	`<hostname>`	`rhxhost01`	`rhxhost02`
Host IP	`<host_ip>`	`10.1.10.1`	`10.1.20.1`
Host additional IP	`<host_additional_ip>`	`10.1.10.2`	`10.1.20.2`
Configured subnet	`<subnet_id>`	`subnet-xxxxxxxxxxsubnet1`	`subnet-xxxxxxxxxxsubnet2`

- Hostname – Hostnames must comply with SAP requirements outlined in {https---launchpad-support-sap-com---notes-611361}[SAP Note 611361 - Hostnames of SAP ABAP Platform servers] (requires SAP portal access).

  Run the following command on your instances to retrieve the hostname.

  ```
  hostname
  ```

- Amazon EC2 instance ID – run the following command (IMDSv2 compatible) on your instances to retrieve instance metadata.

  ```
  /usr/bin/curl --noproxy '*' -w "\n" -s -H "X-aws-ec2-metadata-token: $(curl --noproxy
   '*' -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-
  token-ttl-seconds: 21600")" http://169.254.169.254/latest/meta-data/instance-id
  ```

  For more details, see Retrieve instance metadata and Instance identity documents.

SAP and Pacemaker resource parameters

Name	Parameter	Example
SID	`<SID>` or `<sid>`	`RHX`
ASCS Alias	`<ascs_virt_hostname>`	`rhxascs`
ASCS System Number	`<ascs_sys_nr>`	`00`
ASCS Overlay IP	`<ascs_oip>`	`172.16.30.5`
ASCS NFS Mount Point	`<ascs_nfs_mount_point>`	`/RHX_ASCS00`
ERS Alias	`<ers_virt_hostname>`	`rhxers`
ERS System Number	`<ers_sys_nr>`	`10`
ERS Overlay IP	`<ers_oip>`	`172.16.30.6`
ERS NFS Mount Point	`<ers_nfs_mount_point>`	`/RHX_ERS10`
ENSA Type	`<ensa_type>`	`ENSA2/ENSA1`
VPC Route Tables	`<rtb_id>`	`rtb-xxxxxroutetable1`
Sapmnt NFS ID or CNAME	`<sapmnt_nfs_id>`	`fs-xxxxxxxxxxxxxefs1`

- SAP details – SAP parameters, including SID and instance number must follow the guidance and limitations of SAP and Software Provisioning Manager. Refer to {https---launchpad-support-sap-com---notes-1979280}[SAP Note 1979280 - Reserved SAP System Identifiers (SAPSID) with Software Provisioning Manager] for more details.

  Post-installation, use the following command to find the details of the instances running on a host.

  ```
  sudo /usr/sap/hostctrl/exe/saphostctrl -function ListInstances
  ```

- Overlay IP – This value is defined by you. For more information, see {https---docs-aws-amazon-com-sap-latest-sap-netweaver-rhel-netweaver-ha-planning-html-overlay-ip}[Overlay IP].
- NFS mount points – This value is defined by you. Consider which systems are going to share an NFS file system (Amazon EFS or Amazon FSx), and ensure that your naming standards allow it.

RHEL cluster parameters

Name	Parameter	Example
Cluster name	`cluster_name`	`rhelha`
Cluster user	`cluster_user`	`hacluster`
Cluster password	`cluster_password`	
Cluster connector	`cluster_connector`	`sap-cluster-connector`

Amazon EC2 instance settings

Amazon EC2 instance settings can be applied using Infrastructure as Code or manually using AWS Command Line Interface or AWS Management Console. We recommend Infrastructure as Code automation to reduce manual steps, and ensure consistency.

Topics

Create IAM roles and policies

In addition to the permissions required for standard SAP operations, two IAM policies are required for the cluster to control AWS resources on ASCS. These policies must be assigned to your Amazon

EC2 instance using an IAM role. This enables Amazon EC2 instance, and therefore the cluster to call AWS services.

Create these policies with least-privilege permissions, granting access to only the specific resources that are required within the cluster. For multiple clusters, you need to create multiple policies.

For more information, see {https---docs-aws-amazon-com-AWSEC2-latest-UserGuide-iam-roles-for-amazon-ec2-html-ec2-instance-profile}[IAM roles for Amazon EC2].

STONITH policy

The RHEL STONITH agent requires permission to start and stop both the nodes of the cluster. Create a policy as shown in the following example. Attach this policy to the IAM role assigned to both Amazon EC2 instances in the cluster.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": [
              "arn:aws:ec2:<region>:<account_id>:instance/<instance_id_1>",
              "arn:aws:ec2:<region>:<account_id>:instance/<instance_id_2>"
            ]
        }
    ]
}
```

AWS Overlay IP policy

The RHEL Overlay IP resource agent (`aws-vpc-move-ip`) requires permission to modify a routing entry in route tables. Create a policy as shown in the following example. Attach this policy to the IAM role assigned to both Amazon EC2 instances in the cluster.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:ReplaceRoute",
            "Resource": [
                 "arn:aws:ec2:<region>:<account_id>:route-table/<rtb_id_1>",
                 "arn:aws:ec2:<region>:<account_id>:route-table/<rtb_id_2>"
                        ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeRouteTables",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> If you are using a Shared VPC, see Shared VPC – optional.

Assign IAM role

The two cluster resource IAM policies must be assigned to an IAM role associated with your Amazon EC2 instance. If an IAM role is not associated to your instance, create a new IAM role for cluster operations. To assign the role, go to https://console.aws.amazon.com/ec2/, select each or both instance(s), and then choose **Actions** > **Security** > **Modify IAM role**.

Modify security groups for cluster communication

A security group controls the traffic that is allowed to reach and leave the resources that it is associated with. For more information, see Control traffic to your AWS resources using security groups.

In addition to the standard ports required to access SAP and administrative functions, the following rules must be applied to the security groups assigned to both Amazon EC2 instances in the cluster.

Inbound

Source	Protocol	Port range	Description
The security group ID (its own resource ID)	**UDP**	5405	Allows UDP traffic between cluster resources for corosync communication

Note

Note the use of the UDP protocol.

If you are running a local firewall, such as `iptables`, ensure that communication on the preceding ports is allowed between two Amazon EC2 instances.

Disable source/destination check

Amazon EC2 instances perform source/destination checks by default, requiring that an instance is either the source or the destination of any traffic it sends or receives.

In the pacemaker cluster, source/destination check must be disabled on both instances receiving traffic from the Overlay IP. You can disable check using AWS CLI or AWS Management Console.

AWS CLI

- Use the modify-instance-attribute command to disable source/destination check.

  Run the following commands on both instances in the cluster.

  - Primary example –

```
aws ec2 modify-instance-attribute --instance-id <i-xxxxinstidforhost1> --no-
source-dest-check
```

- Secondary example –

```
aws ec2 modify-instance-attribute --instance-id <i-xxxxinstidforhost2> --no-
source-dest-check
```

AWS Management Console

- Ensure that the **Stop** option is checked in https://console.aws.amazon.com/ec2/.

Review automatic recovery and stop protection

After a failure, cluster-controlled operations must be resumed in a coordinated way. This helps ensure that the cause of failure is known and addressed, and the status of the cluster is as expected. For example, verifying that there are no pending fencing actions.

This can be achieved by not enabling pacemaker to run as a service at the operating system level or by avoiding auto restarts for hardware failure.

If you want to control the restarts resulting from hardware failure, disable simplified automatic recovery and do not configure Amazon CloudWatch action-based recovery for Amazon EC2 instances that are part of a pacemaker cluster. Use the following commands on both Amazon EC2 instances in the pacemaker cluster, to disable simplified automatic recovery via AWS CLI. If making the change via AWS CLI, run the command for both Amazon EC2 instances in the cluster.

> **Note**
>
> Modifying instance maintenance options will require admin privileges not covered by the IAM instance roles defined for operations of the cluster.

```
aws ec2 modify-instance-maintenance-options --instance-id <i-xxxxinstidforhost1> --
auto-recovery disabled
```

```
aws ec2 modify-instance-maintenance-options --instance-id <i-xxxxinstidforhost2> --
auto-recovery disabled
```

To ensure that STONITH actions can be executed, you must ensure that stop protection is disabled for Amazon EC2 instances that are part of a pacemaker cluster. If the default settings have been modified, use the following commands for both instances to disable stop protection via AWS CLI.

Note

Modifying instance attributes will require admin privileges not covered by the IAM instance roles defined for operations of the cluster.

```
aws ec2 modify-instance-attribute --instance-id <i-xxxxinstidforhost1> --no-disable-
api-stop
```

```
aws ec2 modify-instance-attribute --instance-id <i-xxxxinstidforhost2> --no-disable-
api-stop
```

Operating system prerequisites

This section covers the following topics.

Topics

Root access

Verify root access on both cluster nodes. The majority of the setup commands in this document are performed with the root user. Assume that commands should be run as root unless there is an explicit call out to choose otherwise.

Install missing operating system packages

This is applicable to both cluster nodes. You must install any missing operating system packages.

The following packages and their dependencies are required for the pacemaker setup. Depending on your baseline image, for example, RHEL for SAP, these packages may already be installed.

```
awscli
chrony
corosync
pcs
pacemaker
fence-agents-aws
resource-agents-sap (Version resource-agents-sap-3.9.5-124.el7.x86_64 or higher)
sap-cluster-connector
```

We highly recommend installing the following additional packages for troubleshooting.

```
sysstat
pcp-system-tools
sos
```

See Red Hat documentation What are all the Performance Co-Pilot (PCP) RPM packages in RHEL?

> **Note**
>
> The preceding list of packages is not a complete list required for running SAP applications. For the complete list, see {https---docs-aws-amazon-com-sap-latest-sap-netweaver-rhel-netweaver-ha-planning-html-references}[SAP and Red Hat references].

Use the following command to check packages and versions.

```
for package in awscli chrony corosync pcs pacemaker fence-agents-aws resource-agents-
sap sap-cluster-connector sysstat pcp-system-tools sos; do
echo "Checking if ${package} is installed..."
RPM_RC=$(rpm -q ${package} --quiet; echo $?)
if [ ${RPM_RC} -ne 0 ];then
echo "   ${package} is missing and needs to be installed"
fi
done
```

If a package is not installed, and you are unable to install it using `yum`, it may be because Red Hat Enterprise Linux for SAP extension is not available as a repository in your chosen image. You can verify the availability of the extension using the following command.

```
yum repolist
```

To install or update a package or packages with confirmation, use the following command.

```
yum install <package_name(s)>
```

Update and check operating system versions

You must update and confirm versions across nodes. Apply all the latest patches to your operating system versions. This ensures that bugs are addresses and new features are available.

You can update the patches individually or use the `yum` update. A clean reboot is recommended prior to setting up a cluster.

```
yum update
reboot
```

Compare the operating system package versions on the two cluster nodes and ensure that the versions match on both nodes.

Stop and disable `nm-cloud-setup`

This is applicable on both cluster nodes. If you are using Red Hat 8.6 or later, the following services must be stopped and disabled on both the cluster nodes. This prevents the NetworkManager from removing the overlay IP address from the network interface.

```
systemctl disable nm-cloud-setup.timer
systemctl stop nm-cloud-setup.timer
systemctl disable nm-cloud-setup
systemctl stop nm-cloud-setup
```

Time synchronization services

This is applicable to both cluster nodes. Time synchronization is important for cluster operation. Ensure that `chrony rpm` is installed, and configure appropriate time servers in the configuration file.

You can use Amazon Time Sync Service that is available on any instance running in a VPC. It does not require internet access. To ensure consistency in the handling of leap seconds, don't mix Amazon Time Sync Service with any other `ntp` time sync servers or pools.

Create or check the `/etc/chrony.d/ec2.conf` file to define the server.

```
Amazon EC2 time source config
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

Start the `chronyd.service`, using the following command.

```
systemctl enable --now chronyd.service
systemctl status chronyd
```

For more information, see Set the time for your Linux instance.

AWS CLI profile

This is applicable to both cluster nodes. The cluster resource agents use AWS Command Line Interface (AWS CLI). You need to create an AWS CLI profile for the root account on both instances.

You can either edit the config file at `/root/.aws` manually or by using aws configure AWS CLI command.

You can skip providing the information for the access and secret access keys. The permissions are provided through IAM roles attached to Amazon EC2 instances.

```
aws configure
{aws} Access Key ID [None]:
{aws} Secret Access Key [None]:
Default region name [None]: <region_id>
Default output format [None]:
```

Pacemaker proxy settings

This is applicable to both cluster nodes. If your Amazon EC2 instance has been configured to access the internet and/or AWS Cloud through proxy servers, then you need to replicate the settings in the pacemaker configuration. For more information, see Use an HTTP proxy.

Add the following lines to `/etc/sysconfig/pacemaker`.

```
http_proxy=http://<proxyhost>:<proxyport>
https_proxy= http://<proxyhost>:<proxyport>
no_proxy=127.0.0.1,localhost,169.254.169.254,fd00:ec2::254
```

Modify `proxyhost` and `proxyport` to match your settings. Ensure that you exempt the address used to access the instance metadata. Configure `no_proxy` to include the IP address of the instance metadata service – **`169.254.169.254`** (IPV4) and **`fd00:ec2::254`** (IPV6). This address does not vary.

IP and hostname resolution prerequisites

This section covers the following topics.

Topics

- Add initial VPC route table entries for overlay IPs
- Add overlay IPs to host IP configuration
- Hostname resolution

Add initial VPC route table entries for overlay IPs

You need to add initial route table entries for overlay IPs. For more information on overlay IP, see {https---docs-aws-amazon-com-sap-latest-sap-netweaver-rhel-netweaver-ha-planning-html-overlay-ip}[Overlay IP].

Add entries to the VPC route table or tables associated with the subnets of your Amazon EC2 instance for the cluster. The entries for destination (overlay IP CIDR) and target (Amazon EC2 instance or ENI) must be added manually for ASCS and ERS. This ensures that the cluster resource has a route to modify. It also supports the install of SAP using the virtual names associated with the overlay IP before the configuration of the cluster.

Modify or add a route to a route table using AWS Management Console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Route Tables**, and select the route table associated with the subnets where your instances have been deployed.
3. Choose **Actions**, **Edit routes**.
4. To add a route, choose **Add route**. You must choose **Add route** twice to add two routes, one for ASCS and another for ERS.
   a. Add your chosen overlay IP address CIDR and the instance ID of your primary instance for ASCS. See the following table for an **example**.

Destination	172.16.30.5/32

Target	i-xxxxinstidforhost1

b. Add your chosen overlay IP address CIDR and the instance ID of your secondary instance for ERS. See the following table for an **example**.

Destination	172.16.30.6/32
Target	i-xxxxinstidforhost2

5. Choose **Save changes**.

   You route table now has two entries, each for ASCS and ERS, in addition to the standard routes. The selected instances IDs resolve to the corresponding primary Elastic Network Interface (ENI).

The preceding steps can also be performed programmatically. We suggest performing the steps using administrative privileges, instead of instance-based privileges to preserve least privilege. CreateRoute API isn't necessary for ongoing operations.

Run the following command as a dry run on both nodes to confirm that the instances have the necessary permissions.

```
aws ec2 replace-route --route-table-id <rtb-xxxxxroutetable1> --destination-cidr-
block <172.16.30.5/32> --instance-id <i-xxxxinstidforhost1> --dry-run --profile
 <aws_cli_cluster_profile>
```

Add overlay IPs to host IP configuration

You must configure the overlay IP as an additional IP address on the standard interface to enable SAP install. This action is managed by the cluster IP resource. However, to install SAP using the correct IP addresses prior to having the cluster configuration in place, you need to add these entries manually.

If you need to reboot the instance during setup, the assignment is lost, and must be re-added.

See the following **examples**. You must update the commands with your chosen IP addresses.

On EC2 instance 1, where you are installing ASCS, add the overlay IP allocated for ASCS.

```
ip addr add <172.16.30.5/32> dev eth0
```

On EC2 instance 2, where you are installing ERS, add the overlay IP allocated for ERS.

```
ip addr add <172.16.30.6/3> dev eth0
```

Hostname resolution

This is applicable to both cluster nodes. You must ensure that both instances can resolve all hostnames in use. Add the hostnames for cluster nodes to `/etc/hosts` file on both cluster nodes. This ensures that hostnames for cluster nodes can be resolved even in case of DNS issues. See the following example.

```
cat /etc/hosts
<10.1.10.1 rhxhost01.example.com rhxhost01>
<10.1.20.1 rhxhost02.example.com rhxhost02>
<172.16.30.5 rhxascs.example.com rhxascs>
<172.16.30.6 rhxers.example.com rhxers>
```

⚠ **Important**

The overlay IP is out of VPC range, and cannot be reached from locations not associated with the route table, including on-premises.

File system prerequisites

This section covers the following topics.

Topics

- Shared file systems
- Create file systems
- Update /etc/fstab
- Temporarily mount ASCS and ERS directories for installation (classic only)

Shared file systems

Amazon Elastic File System and Amazon FSx for NetApp ONTAP support SAP NetWeaver file systems on Linux. For more information, see Getting started with Amazon Elastic File System and

Getting started with Amazon FSx for NetApp ONTAP. You can evaluate both of these options, and select one based on resilience, performance, and cost.

We recommend sharing a single Amazon EFS or FSx for ONTAP file system across multiple SIDs within an account.

The file system's DNS name is the simplest mounting option. The DNS automatically resolves to the mount target's IP address in the Availability Zone of the connecting Amazon EC2 instance. You can also create an alias to help with identifying the purpose of the shared file system. We have used `<nfs.fqdn>` in this document. The following are some examples.

- file-system-id.efs.aws-region.amazonaws.com
- svm-id.fs-id.fsx.aws-region.amazonaws.com
- qas_sapmnt_share.example.com

Note

Review the `enableDnsHostnames` and `enableDnsSupport` DNS attributes for your VPC. For more information, see {https---docs-aws-amazon-com-vpc-latest-userguide-vpc-dns-html-vpc-dns-updating}[View and update DNS attributes for your VPC].

Create file systems

The following shared file systems are covered in this document.

Unique NFS Location (example)	File system location
RHX_sapmnt	/sapmnt/RHX
RHX_ASCS00	/usr/sap/RHX/ASCS00
RHX_ERS10	/usr/sap/RHX/ERS10

For more information, see SAP Documentation – {https---help-sap-com-docs-SAP-NETWEAVER-750-ff18034f08af4d7bb33894c2047c3b71-2744f17a26a74a8abfd202c4f5dc9a0f-html}[SAP System Directories on UNIX].

The following options can differ depending on how you architect and operate your systems.

- `/usr/sap/trans` is not listed as a required file system for ASCS. You can add this to your setup.
- A shared home directory has not been included. This enables you to log on locally as the `<sid>adm` user, in the event of NFS issues. A shared home directory can be the suitable option if your administrators have root access.

Using the NFS ID created in the previous step, temporarily mount the root directory of the NFS with the following command. `/mnt` is available by default; it can also be substituted with another temporary location.

```
mount <nfs.fqdn>.amazonaws.com:/ /mnt
```

Create the directories using the following command.

```
mkdir -p /mnt/<RHX_sapmnt>
mkdir -p /mnt/<RHX_ASCS00>
mkdir -p /mnt/<RHX_ERS10>
```

Change the ownership or permissions to ensure that the install as `<sid>adm` can write to the directories. If `<sid>adm` is going to be created by the installer, temporarily change the permissions to 777, as seen in the following command.

```
chmod 777 /mnt/<RHX_sapmnt> /mnt/<RHX_ASCS00> /mnt/<RHX_ERS10>
```

Unmount the temporary mount using the following command.

```
umount /mnt
```

Update `/etc/fstab`

This is applicable to both cluster nodes. `/etc/fstab` is a configuration table containing the details required for mounting and unmounting file systems to a host.

Create the directories for the required mount points (permanent or cluster controlled), using the following commands.

```
mkdir /sapmnt
mkdir /usr/sap/<RHX/ASCS00>
mkdir /usr/sap/<RHX/ERS10>
```

Add the file systems not managed by the cluster to `/etc/fstab`.

Prepare and append an entry for the `sapmnt` file system to `/etc/fstab`.

```
<nfs.fqdn>/<RHX_sapmnt>  /sapmnt   nfs
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 0 0
```

Review the mount options to ensure that they match with your operating system, NFS file system type, and latest recommendations from SAP.

Use the following command to mount the file systems defined in `/etc/fstab`.

```
mount -a
```

Use the following command to check that the required file systems are available.

```
df -h
```

Temporarily mount ASCS and ERS directories for installation (classic only)

This is only applicable to the classic architecture. Simple-mount architecture has these directories permanently available in `/etc/fstab`.

You must temporarily mount ASCS and ERS directories for installation.

Use the following command on the instance where you to plan to install ASCS.

```
mount <nfs.fqdn>:/<RHX_ASCS00>  /usr/sap/<RHX>/ASCS
```

Use the following command on the instance where you to plan to install ERS.

```
mount <nfs.fqdn>:/<RHX_ERS10>  /usr/sap/<RHX>/ERS
```

Shared VPC – *optional*

Amazon VPC sharing enables you to share subnets with other AWS accounts within the same AWS Organizations. Amazon EC2 instances can be deployed using the subnets of the shared Amazon VPC.

In the pacemaker cluster, the `aws-vpc-move-ip` resource agent has been enhanced to support a shared VPC setup while maintaining backward compatibility with previous existing features.

The following checks and changes are required. We refer to the AWS account that owns Amazon VPC as the sharing VPC account, and to the consumer account where the cluster nodes are going to be deployed as the cluster account.

This section covers the following topics.

Topics

Minimum version requirements

The latest version of the `aws-vpc-move-ip` agent shipped with Red Hat 8.2 supports the shared VPC setup by default. The following are the minimum version required to support a shared VPC Setup:

- Red Hat 7.9 - resource-agents-4.1.1-61.10
- Red Hat 8.1 - resource-agents-4.1.1-33.10
- Red Hat 8.2 - resource-agents-4.1.1-44.12

IAM roles and policies

Using the overlay IP agent with a shared Amazon VPC requires a different set of IAM permissions to be granted on both AWS accounts (sharing VPC account and cluster account).

Sharing VPC account

In sharing VPC account, create an IAM role to delegate permissions to the EC2 instances that will be part of the cluster. During the IAM Role creation, select "Another AWS account" as the type of trusted entity, and enter the AWS account ID where the EC2 instances will be deployed/running from.

After the IAM role has been created, create the following IAM policy on the sharing VPC account, and attach it to an IAM role. Add or remove route table entries as needed.

```
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "VisualEditor0",
"Effect": "Allow",
"Action": "ec2:ReplaceRoute",
"Resource": [
"arn:aws:ec2:<region>:<sharing_vpc_account_id>:route_table/<rtb_id_1>",
"arn:aws:ec2:<region>:<sharing_vpc_account_id>:route_table/<rtb_id_2>"
]
},
{
"Sid": "VisualEditor1",
"Effect": "Allow",
"Action": "ec2:DescribeRouteTables",
"Resource": "*"
}
]
}
```

Next, edit move to the "Trust relationships" tab in the IAM role, and ensure that the AWS account you entered while creating the role has been correctly added.

Cluster account

In cluster account, create the following IAM policy, and attach it to an IAM role. This is the IAM Role that is going to be attached to the EC2 instances.

AWS STS policy

```
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "VisualEditor0",
"Effect": "Allow",
"Action": "sts:AssumeRole",
"Resource": "arn:aws:iam::<sharing_vpc_account_id>:role/<sharing _vpc-account-cluster-role>"
}
]
}
```

STONITH policy

```
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "VisualEditor0",
"Effect": "Allow",
"Action": [
"ec2:StartInstances",
"ec2:StopInstances"
],
"Resource": [
"arn:aws:ec2:<region>:<cluster_account_id>:instance/<instance_id_1>",
"arn:aws:ec2:<region>:<cluster_account_id>:instance/<instance_id_2>"
]
},
{
"Sid": "VisualEditor1",
"Effect": "Allow",
"Action": "ec2:DescribeInstances",
"Resource": "*"
}
]
```

Shared VPC cluster resources

The cluster resource agent `aws-vpc-move-ip` also uses a different configuration syntax. When configuring the `aws-vpc-move-ip` resource agent, the following new parameters must be used:

- lookup_type=NetworkInterfaceId
- routing_table_role="arn:aws:iam::<account_id>:role/<VPC-Account-Cluster-Role>"

The following IP Resource for ASCS needs to be created.

```
pcs resource create <rsc_ip_RHX_ASCS00> ocf:heartbeat:aws-vpc-move-ip ip=172.16.30.5
 interface=eth0  routing_table=<rtb-xxxxxroutetable1> lookup_type=NetworkInterfaceId
  routing_table_role="<arn:aws:iam:<sharing_vpc_account_id>:role/
<sharing_vpc_account_cluster_role>>" op monitor interval=20s timeout=40s --group
 <grp_RHX_ASCS00>
```

The following IP Resource for ERS needs to be created.

```
pcs resource create <rsc_ip_RHX_ERS10> ocf:heartbeat:aws-vpc-move-ip
 params ip=172.16.30.6 interface=eth0  routing_table=<rtb-xxxxxroutetable1>
 lookup_type=NetworkInterfaceId
 routing_table_role="<arn:aws:iam::<sharing_vpc_account_id>:role/
<sharing_vpc_account_cluster_role>>" op monitor interval=20s timeout=40s --group
 <grp_RHX_ERS10>
```

SAP and cluster setup

This section covers the following topics.

Topics

- Install SAP
- Modify SAP control operations for cluster use
- Cluster prerequisites
- Create cluster and node associations

Install SAP

The following topics provide information about installing SAP on AWS Cloud in a highly available cluster. Review SAP Documentation for more details.

Topics

- Use SWPM with high availability
- Install SAP instances
- Kernel upgrade and ENSA2 – optional
- Check SAP host agent version

Use SWPM with high availability

Before running SAP Software Provisioning Manager (SWPM), ensure that the following prerequisites are met.

- If the operating system groups for SAP are pre-defined, ensure that the user identifier (UID) and group identifier values for `<sid>adm` and `sapsys` are consistent across both instances.
- You have downloaded the most recent version of Software Provisioning Manager for your SAP version. For more information, see SAP Documentation Software Provisioning Manager.
- Ensure that routes, overlay IPs, and virtual host names are mapped to both instances. This is to ensure that the virtual hostname for ASCS is available on instance 1, and the virtual hostname for ERS is available on instance 2. For more information, see IP and hostname resolution prerequisites.
- Ensure that shared file systems are available, either in `/etc/fstab` or using the mount command. For more information, see File system prerequisites.

Install SAP instances

The commands in this section use the example values provided in Define reference parameters for setup.

Install ASCS instance on `<rhxhost01>` with virtual hostname `rhxascs`, using the high availability option of Software Provisioning Manager (SWPM) tool. You can use the `SAPINST_USE_HOSTNAME` parameter to install SAP using a virtual hostname.

```
<swpm location>/sapinst SAPINST_USE_HOSTNAME=<rhxascs>
```

Install ERS instance on `<rhxhost02>` with virtual hostname `rhxers`, using the high availability option of Software Provisioning Manager (SWPM) tool. You can use the `SAPINST_USE_HOSTNAME` parameter to install SAP using a virtual hostname.

```
<swpm location>/sapinst SAPINST_USE_HOSTNAME=<rhxers>
```

Once installation is complete, install and configure the database and SAP Primary Application Server (PAS). Optionally, you can also install and configure Additional Application Server (AAS). For more details on installing SAP NetWeaver, refer SAP Help Portal.

For additional information on unattended installation options, see SAP Note 2230669 – System Provisioning Using an Input Parameter File (requires SAP portal access).

Kernel upgrade and ENSA2 – *optional*

As of AS ABAP Release 7.53 (ABAP Platform 1809), the new Standalone Enqueue Server 2 (ENSA2) is installed by default. ENSA2 replaces the previous version – ENSA1.

If you have an older version of SAP NetWeaver, consider following the SAP guidance to upgrade the kernel and update the Enqueue Server configuration. An upgrade will allow you to take advantage of the features available in the latest version. For more information, see the following SAP Notes (require SAP portal access).

- SAP Note 2630416 – Support for Standalone Enqueue Server 2
- SAP Note 2711036 – Usage of the Standalone Enqueue Server 2 in an HA Environment

Check SAP host agent version

This is applicable to both cluster nodes. The SAP host agent is used for system instance control and monitoring. This agent is used by SAP cluster resource agents and hooks. It is recommended that you have the latest version installed on both instances. For more details, see SAP Note 2219592 – Upgrade Strategy of SAP Host Agent.

Use the following command to check the version of the host agent.

```
/usr/sap/hostctrl/exe/saphostexec -version
```

Modify SAP control operations for cluster use

This section covers the following topics.

Topics

Add `sidadm` to `haclient` group

This is applicable to both cluster nodes. An `haclient` operating system group is created when the cluster connector package is installed. Adding the `sidadm` user to this group ensures that your cluster has necessary access. Run the following command as root.

```
usermod -a -G haclient <rhx>adm
```

Modify SAP profiles for start operations and cluster hook

This action ensures that there is compatibility between SAP start framework and cluster actions. Modify SAP profiles to change the start behavior of the SAP instance and processes. Ensure that `sapcontrol` is aware that the system is being managed by a pacemaker cluster.

The following changes must be made in the instance profiles for ASCS and ERS. These profiles are created during install, and are located at `/usr/sap/<SID>/SYS/profile/`.

- ASCS profile example – `/usr/sap/RHX/SYS/profile/RHX_ASCS00_rhxascs`
- ERS profile example – `/usr/sap/RHX/SYS/profile/RHX_ERS10_rhxers`

1. **Program or process start behavior –** In case of failure, processes must be restarted. Determining where the process starts and in what order needs to be controlled by the cluster, and not SAP start framework behavior defined in the profiles. Your locks can be lost if this parameter is not changed.

   Example

   ENSA1

   ASCS

   ```
   #For ENSA1 (_EN)
   #Changing Restart to Start for Cluster compatibility
   #Old value: Restart_Program_XX = local $(_EN) pf=$(_PF)

   Start_Program_XX = local $(_EN) pf=$(_PF)
   ```

   ERS

   ```
   #For ENSA1 (_ER)
   #Changing Restart to Start for Cluster compatibility
   ```

```
#Old value: Restart_Program_XX = local $(_ER) pf=$(_PF)NR=$(SCSID)

Start_Program_XX = local $(_ER) pf=$(_PF) NR=$(SCSID)
```

XX *indicates the start-up order. This value may be different in your install; retain the unchanged value.*

ENSA2

ASCS

```
#For ENSA2 (_ENQ)
#Changing Restart to Start for Cluster compatibility
#Old value: Restart_Program_XX = local $(_ENQ) pf=$(_PF)

Start_Program_XX = local $(_ENQ) pf=$(_PF)
```

ERS

```
#For ENSA2 (_ENQR)
#Changing Restart to Start for Cluster compatibility
#Old value: Restart_Program_XX = local $(_ENQR) pf=$(_PF)NR=$(SCSID)

Start_Program_XX = local $(_ENQR) pf=$(_PF) NR=$(SCSID)
```

XX *indicates the start order. This value may be different in your install; retain the unchanged value.*

2. **Disable instance auto start in both profiles –** When an instance restarts, SAP start framework should not start ASCS and ERS automatically. Add the following parameter on both profiles to prevent an auto start.

```
Autostart = 0
```

3. **Add cluster connector details in both profiles –** The connector integrates the SAP start and control frameworks of SAP NetWeaver with RHEL cluster to assist with maintenance and awareness of state. Add the following parameters on both profiles.

```
Added for Cluster Connectivity
service/halib = $(DIR_CT_RUN)/saphascriptco.so
service/halib_cluster_connector = /usr/bin/sap_cluster_connector
```

> ⚠ **Important**
>
> The minimum version of `sap_cluster_connector` that complies with HA-Interface certification NW-HA-CLU 750 or S/4-HA-CLU 1.0 is 3.0.1-1.el7_6.5. A previous version of SAP cluster connector was delivered as part of the `resource-agents-sap` package and the name of the connector was `sap_cluster_connector`.

4. **Restart services** – Restart SAP services for ASCS and ERS to ensure that the preceding settings take effect. Adjust the system number to match the service.

   ASCS

   ```
   /usr/sap/hostctrl/exe/sapcontrol -nr <00> -function RestartService
   ```

   ERS

   ```
   /usr/sap/hostctrl/exe/sapcontrol -nr <10> -function RestartService
   ```

5. **Check integration using `sapcontrol`** – `sapcontrol` includes two functions: `HACheckConfig` and `HACheckFailoverConfig`. These functions can be used to check configuration, including awareness of the cluster connector.

   ASCS

   ```
   /usr/sap/hostctrl/exe/sapcontrol -nr <00> -function HACheckFailoverConfig
   /usr/sap/hostctrl/exe/sapcontrol -nr <00> -function HACheckConfig
   ```

   ERS

   ```
   /usr/sap/hostctrl/exe/sapcontrol -nr 10 -function HACheckFailoverConfig
   /usr/sap/hostctrl/exe/sapcontrol -nr 10 -function HACheckConfig
   ```

Modify `sapservices`

This is applicable to both cluster nodes. In older versions of Red Hat and SAP kernel, the `systemV init` service `sapinit` is responsible for starting SAP host agent and all `sapstartsrv` processes listed in `/usr/sap/sapservices`. In newer versions, native integration is available between systemd and SAP services. For more details, see the following SAP Notes (require SAP portal access).

- SAP Note 3139184 – Linux: systemd integration for sapstartsrv and SAP Host Agent
- SAP Note 3115048 – sapstartsrv with native Linux systemd support

Review if systemV or systemd integration is in place, and is consistent for ASCS and ERS by checking the /usr/sap/services file on both nodes.

```
cat /usr/sap/sapservices
```

See the following table for more details.

Example

systemV

See the following example ASCS entry for older version with systemV integration.

```
LD_LIBRARY_PATH=/usr/sap/<RHX>/ASCS<00>/exe:$LD_LIBRARY_PATH;export
 LD_LIBRARY_PATH;/usr/sap/<RHX>/ASCS<00>/exe/sapstartsrv pf=/usr/sap/<RHX>/SYS/
profile/<RHX>_ASCS<00>_<rhxascs> -D -u <rhxadm>u
```

To ensure that SAP instance can be managed by the cluster and also manually during planned maintenance activities, add the missing entries for ASCS and ERS sapstartsrv service in /usr/sap/sapservices file on both cluster nodes (ASCS and ERS host). Copy the missing entry from both hosts. Post-modifications, the /usr/sap/sapservices file looks as follows on both hosts.

```
#!/bin/sh
LD_LIBRARY_PATH=/usr/sap/<RHX>/ASCS<00>/exe:$LD_LIBRARY_PATH; export
 LD_LIBRARY_PATH; /usr/sap/<RHX>/ASCS<00>/exe/sapstartsrv pf=/usr/sap/<RHX>/SYS/
profile/<RHX>_ASCS<00>_<rhxascs> -D -u <rhxadm>

LD_LIBRARY_PATH=/usr/sap/<RHX>/ERS<10>/exe:$LD_LIBRARY_PATH; export
 LD_LIBRARY_PATH; /usr/sap/<RHX>/ERS<10>/exe/sapstartsrv pf=/usr/sap/<RHX>/SYS/
profile/<RHX>_ERS<10>_ -D<rhxers> -u <rhxadm>
```

systemd

- See the following example ASCS entry for newer version with native `systemd` integration.

  ```
  systemctl --no-ask-password start <SAPRHX_00> # sapstartsrv pf=/usr/sap/<RHX>/SYS/
  profile/<RHX>_ASCS<00>_<rhxascs>
  ```

  To ensure that SAP instance can be managed by the cluster and also manually during planned maintenance activities, add the missing entries for ASCS and ERS `sapstartsrv` service in `/usr/sap/sapservices` file on both cluster nodes (ASCS and ERS host). Copy the missing entry from both hosts. Post-modifications, the `/usr/sap/sapservices` file looks as follows on both hosts.

  ```
  #!/bin/sh
  systemctl --no-ask-password start <SAPRHX_00> # sapstartsrv pf=/usr/sap/<RHX>/SYS/
  profile/<RHX>_ASCS<00>_<rhxascs>

  systemctl --no-ask-password start <SAPRHX_10> # sapstartsrv pf=/usr/sap/<RHX>/SYS/
  profile/<RHX>_ERS<10>_<rhxers>
  ```

Align and disable SAP auto start services for `systemd`

This is applicable to both cluster nodes. For `systemd`, ensure SAP auto start services are aligned and disabled across nodes. If the installed version supports native integration with `systemd`, you must create services for ASCS and ERS on both nodes. This ensures that if you have revert to manual operations, there is no association and that both nodes are configured in the same manner.

You must disable auto start services to enable the cluster to manage stop/start.

ASCS

Register the missing ERS service on the node where you have installed ASCS.

Temporarily mount the ERS directory.

```
mount <nfs.fqdn>:/<RHX>_ERS<10>  /usr/sap/<RHX>/ERS

```

Register the ERS service.

```
export LD_LIBRARY_PATH=/usr/sap/<RHX>/ERS<10>/exe
/usr/sap/<RHX>/ERS<10>/exe/sapstartsrv pf=/usr/sap/<RHX>/SYS/profile/
<RHX>_ERS<10>_<rhxers> -reg

systemctl start <SAPRHX_10>
```

Check the existence and state of SAP services.

```
systemctl list-unit-files SAP*
UNIT FILESTATE VENDOR PRESET
<SAPRHX_00>.service disabled disabled
<SAPRHX_10>.service disabled disabled
SAP.slicestatic -
3 unit files listed.
```

If the state is not disabled, run the following command to disable `sapservices` integration for `SAP<SID>_<ascs_sys_nr>` and `SAP<SID>_<ers_sys_nr>` on both nodes.

Important

Stopping these services also stops the associated SAP instances.

```
systemctl stop <SAPRHX_00>.service
systemctl disable <SAPRHX_00>.service
systemctl stop <SAPRHX_10>.service
systemctl disable <SAPRHX_10>.service
```

Unmount the ERS directory (classic only).

```
umount /usr/sap/<RHX>/ERS
```

ERS

Register the missing ASCS service on the node where you have installed ERS.

Temporarily mount the ASCS directory (classic only).

```
mount <nfs.fqdn>:/<RHX>_ASCS<00>  /usr/sap/<RHX>/ASCS
```

Register the ASCS service.

```
export LD_LIBRARY_PATH=/usr/sap/<RHX>/ASCS<00>/exe
/usr/sap/<RHX>/ASCS<00>/exe/sapstartsrv pf=/usr/sap/<RHX>/SYS/profile/
<RHX>_ASCS<00>_<rhxascs> -reg
systemctl start <SAPRHX_00>
```

Check the existence and state of SAP services.

```
systemctl list-unit-files SAP*
UNIT FILESTATE VENDOR PRESET
<SAPRHX_00>.service disabled disabled
<SAPRHX_10>.service disabled disabled
SAP.slicestatic -
3 unit files listed.
```

If the state is not disabled, run the following command to disable `sapservices` integration for `SAP<SID>_<ascs_sys_nr>` and `SAP<SID>_<ers_sys_nr>` on both nodes.

Important

Stopping these services also stops the associated SAP instances.

```
systemctl stop <SAPRHX_00>.service
systemctl disable <SAPRHX_00>.service
systemctl stop <SAPRHX_10>.service
systemctl disable <SAPRHX_10>.service
```

Unmount the ASCS directory (classic only).

```
umount /usr/sap/<RHX>/ASCS
```

For more details, see SAP Note 3139184 – Linux: systemd integration for sapstartsrv and SAP Host Agent.

Cluster prerequisites

This section covers the following topics.

Topics

- Configure systemd for resource dependencies
- Configure drop-in files for systemd
- Update the hacluster password
- Setup passwordless authentication between nodes

Configure `systemd` for resource dependencies

This is applicable to both cluster nodes. Some failure scenarios, such as an accidental shutdown of an Amazon EC2 instance, can result in unexpected fencing actions. This is caused by pacemaker dependencies that are not directly associated with cluster resources and constraints, but instead are a dependency for the pacemaker service.

If `systemd` is configured, create a config file that defines a dependency between pacemaker and SAP services.

```
mkdir -p /etc/systemd/system/resource-agents-deps.target.d/
cd /etc/systemd/system/resource-agents-deps.target.d/

cat > sap_systemd_<rhx>.conf <<_EOF
[Unit]
Requires=sapinit.service
After=sapinit.service
After=<SAPRHX_00>.service
After=<SAPRHX_10>.service
_EOF


systemctl daemon-reload
```

Configure drop-in files for `systemd`

`systemd` has an inbuilt mechanism to restart a crashed service in a running system. In a cluster-controlled environment, it is recommended to prevent this from getting triggered, so that the cluster can completely manage the respective instances.

This is applicable on both cluster nodes. To prevent the automatic restart of crashed `systemd` based SAP service, create the drop-in files.

```
mkdir -p /etc/systemd/system/<SAPRHX_00>.service.d/
mkdir -p /etc/systemd/system/<SAPRHX_10>.service.d/

cd /etc/systemd/system/<SAPRHX_00>.service.d/

cat > HA.conf <<_EOF
[Service]
Restart=no
_EOF

cd /etc/systemd/system/<SAPRHX_10>.service.d/

cat > HA.conf <<_EOF
[Service]
Restart=no
_EOF

systemctl daemon-reload
```

Update the `hacluster` password

This is applicable to both cluster nodes. Change the password of the operating system user `hacluster` using the following command.

```
passwd <hacluster>
```

Setup passwordless authentication between nodes

For a more comprehensive and easily consumable view of cluster activity, Red Hat provides additional reporting tools. Many of these tools require access to both nodes without entering a password. Red Hat recommends performing this setup for root user.

For more details, see Red Hat documentation How to setup SSH Key passwordless login in Red Hat Enterprise Linux?

Create cluster and node associations

This section covers the following topics.

Topics

- Start pcsd service
- Reset configuration – optional
- Authenticate pcs with user hacluster
- Setup node configuration

Start `pcsd` service

This is applicable on both clsuter nodes. Run the following command to enable and start the cluster service `pcsd` (pacemaker/corosync configuration system daemon) on both, the primary and secondary node.

```
systemctl start pcsd.service
systemctl enable pcsd.service
```

Run the following command to check the status of cluster service.

```
systemctl status pcsd.service
# pcsd.service - PCS GUI and remote configuration interface
   Loaded: loaded (/usr/lib/systemd/system/pcsd.service; enabled; vendor preset:
 disabled)
   Active: active (running) since Fri 2023-01-13 14:15:32 IST; 7min ago
     Docs: man:pcsd(8)
           man:pcs(8)
 Main PID: 1445 (pcsd)
    Tasks: 1 (limit: 47675)
   Memory: 27.1M
   CGroup: /system.slice/pcsd.service
           ##1445 /usr/libexec/platform-python -Es /usr/sbin/pcsd
```

Reset configuration – *optional*

> **Note**
>
> The following instructions help you reset the complete configuration. Run these commands only if you want to start setup from the beginning. You can make minor changes with the `crm edit` command.

Run the following command to back up the current configuration for reference.

```
pcs config show > /tmp/pcsconfig_backup.txt
```

Run the following command to clear the current configuration.

```
pcs cluster destroy
```

Authenticate pcs with user hacluster

The following command authenticates `pcs` to the `pcs daemon` on cluster nodes. It should be run on only one of the cluster nodes. The username and password for the `pcs` user must be the same, and the username should be `hacluster`.

RHEL 7.x

```
pcs cluster auth <rhxhost01> <rhxhost02>
Username: <hacluster>
Password:
<rhxhost02>: Authorized
<rhxhost01>: Authorized
```

RHEL 8.x

```
pcs host auth <rhxhost01> <rhxhost02>
Username: <hacluster>
Password:
<rhxhost02>: Authorized
<rhxhost01>: Authorized
```

Setup node configuration

The following command configures the `cluster configuration` file, and syncs the configuration on both nodes. It should be run on only one of the cluster nodes.

RHEL 7.x

```
pcs cluster setup --name <rhelha> <rhxhost01> <rhxhost02>
Destroying cluster on nodes: <rhxhost01>, <rhxhost02>...
<rhxhost02>: Stopping Cluster (pacemaker)...
<rhxhost01>: Stopping Cluster (pacemaker)...
```

```
<rhxhost02>: Successfully destroyed cluster
<rhxhost01>: Successfully destroyed cluster

Sending 'pacemaker_remote authkey' to '<rhxhost01>', '<rhxhost02>'
<rhxhost01>: successful distribution of the file 'pacemaker_remote authkey'
<rhxhost02>: successful distribution of the file 'pacemaker_remote authkey'
Sending cluster config files to the nodes...
<rhxhost01>: Succeeded
<rhxhost02>: Succeeded

Synchronizing pcsd certificates on nodes <rhxhost01>, <rhxhost02>...
<rhxhost01>: Success
<rhxhost02>: Success
Restarting pcsd on the nodes in order to reload the certificates...
<rhxhost01>: Success
<rhxhost02>: Success.
```

RHEL 8.x

```
#pcs cluster setup <rhelha> <rhxhost01> <rhxhost02>
No addresses specified for host '<rhxhost01>', using '<rhxhost01>'
No addresses specified for host '<rhxhost02>', using '<rhxhost02>'
Destroying cluster on hosts: '<rhxhost01>', '<rhxhost02>'...
<rhxhost01>: Successfully destroyed cluster
<rhxhost02>: Successfully destroyed cluster
Requesting remove 'pcsd settings' from '<rhxhost01>', '<rhxhost02>'
<rhxhost01>: successful removal of the file 'pcsd settings'
<rhxhost02>: successful removal of the file 'pcsd settings'
Sending 'corosync authkey', 'pacemaker authkey' to '<rhxhost01>', '<rhxhost02>'
<rhxhost01>: successful distribution of the file 'corosync authkey'
<rhxhost01>: successful distribution of the file 'pacemaker authkey'
<rhxhost02>: successful distribution of the file 'corosync authkey'
<rhxhost02>: successful distribution of the file 'pacemaker authkey'
Sending 'corosync.conf' to '<rhxhost01>', '<rhxhost02>'
<rhxhost01>: successful distribution of the file 'corosync.conf'
<rhxhost02>: successful distribution of the file 'corosync.conf'
Cluster has been successfully set up.
```

Cluster configuration

This section covers the following topics.

Topics

- Cluster resources
- Sample configuration (`pcs config show`)

Cluster resources

This section covers the following topics.

Topics

- Enable and start the cluster
- Check cluster status
- Prepare for resource creation
- Cluster bootstrap
- Create fence_aws STONITH resource
- Create file system resources
- Create overlay IP resources
- Create SAPInstance resources
- Create resource constraints
- Activate cluster

Enable and start the cluster

This is applicable to both cluster nodes. Run the following command to enable and start the `pacemaker` cluster service on both nodes.

```
pcs cluster enable --all
<rhxhost01>: Cluster Enabled
<rhxhost02>: Cluster Enabled


pcs cluster start --all
<rhxhost01>: Starting Cluster...
<rhxhost02>: Starting Cluster...
```

By enabling the `pacemaker` service, the server automatically joins the cluster after a reboot. This ensures that your system is protected. Alternatively, you can start the `pacemaker` service manually on boot. You can then investigate the cause of failure. However, this is generally not required for SAP NetWeaver ASCS cluster.

Check cluster status

Once the cluster service `pacemaker` is started, check the cluster status with `pcs status` command, as shown in the following example. Both the primary (`rhxhost01`) and secondary (`rhxhost02`) servers should be seen as online.

```
pcs status
Cluster name: <rhelha>

WARNINGS:
No stonith devices and stonith-enabled is not false

Cluster Summary:
  * Stack: corosync
  * Current DC: <rhxhost01> (version <2.0.3-5.el8_2.5-4b1f869f0f>) - partition with
 quorum
  * Last updated: Tue Jan 10 21:32:15 2023
  * Last change:  Tue Jan 10 19:46:50 2023 by hacluster via crmd on <rhxhost01>
  * 2 nodes configured
  * 0 resource instances configured

Node List:
  * Online: [ <rhxhost01> <rhxhost02> ]

Full List of Resources:
  * No resources

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

Prepare for resource creation

To ensure that the cluster does not perform any unexpected actions during setup of resources and configuration, set the maintenance mode to true.

Run the following command to put the cluster in maintenance mode.

```
pcs property set maintenance-mode=true
```

Cluster bootstrap

Configure the cluster bootstrap parameters by running the following commands.

```
pcs resource defaults update resource-stickiness=1
pcs resource defaults update migration-threshold=3
```

Create `fence_aws` STONITH resource

Modify the commands in the following table to match your configuration values.

```
pcs stonith create <rsc_aws_stonith_RHX> fence_aws region=<us-east-1>
 pcmk_host_map="<rhxhost01>:<i-xxxxinstidforhost1>;<rhxhost02>:<i-xxxxinstidforhost2>"
 power_timeout=240 pcmk_reboot_timeout=300 pcmk_reboot_retries=2 pcmk_delay_max=30
 pcmk_reboot_action=reboot op start timeout=180 op stop timeout=180 op monitor
 interval=180 timeout=60
```

Note

The default `pcmk` action is reboot. If you want to have the instance remain in a stopped state until it has been investigated, and then manually started again, add `pcmk_reboot_action=off`. Any High Memory (`u-*tb1.*`) instance or metal instance running on a dedicated host won't support reboot, and will require `pcmk_reboot_action=off`.

Create file system resources

Mounting and unmounting of file system resources to align with the location of the SAP services is done using cluster resources.

Modify and run the following commands to create the file system resources.

ASCS

Use the following command to create ASCS file system resources.

```
pcs resource create <rsc_fs_RHX_ASCS0> ocf:heartbeat:Filesystem
 device=<nfs.fqdn>:/<RHX_ASCS00> directory=</usr/sap/RHX/ASCS00> fstype=nfs4
 options="<nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport>"
 force_unmount=safe --group <grp_RHX_ASCS00>
```

ERS

Use the following command to create ERS file system resources.

```
pcs resource create <rsc_fs_RHX_ERS10> ocf:heartbeat:Filesystem
 device=<nfs.fqdn>:/<RHX_ERS10> directory=</usr/sap/RHX/ERS10> fstype=nfs4
 options="<nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport>"
 force_unmount=safe --group <grp_RHX_ERS10>
```

Notes for ASCS and ERS

- Review the mount options to ensure that they match with your operating system, NFS file system type, and the latest recommendations from SAP.
- <nfs.fqdn> can either be an alias or the default file system resource name of the NFS or FSx for ONTAP resource. For example, `fs-xxxxxx.efs.xxxxxx.amazonaws.com`.

Create overlay IP resources

The IP resource provides the details necessary to update the route table entry for overlay IP.

Modify and run the commands in the table to create IP resources.

ASCS

Use the following command to create an ASCS IP resource.

```
pcs resource create <rsc_ip_RHX_ASCS00> ocf:heartbeat:aws-vpc-move-ip ip=<172.16.30.5>
 interface=eth0 routing_table=<rtb-xxxxxroutetable1> op monitor interval=20s
 timeout=40s --group <grp_RHX_ASCS00>
```

ERS

Use the following command to create an ERS IP resource.

```
pcs resource create <rsc_ip_RHX_ERS10> ocf:heartbeat:aws-vpc-move-ip ip=<172.16.30.6>
 interface=eth0 routing_table=<rtb-xxxxxroutetable1> op monitor interval=20s
 timeout=40s --group <grp_RHX_ERS10>
```

Notes for ASCS and ERS

- If more than one route table is required for connectivity or because of subnet associations, the `routing_table` parameter can have multiple values separated by a comma. For example, `routing_table=rtb-xxxxxroutetable1, rtb-xxxxxroutetable2`.
- Additional parameters – `lookup_type` and `routing_table_role` are required for shared VPC. For more information, see Shared VPC – optional.

Create `SAPInstance` resources

The SAP instance is started and stopped using cluster resources. Modify and run the commands in the table to create `SAPInstance` resources.

> **Note**
>
> The change between ENSA1 and ENSA2 that allows the lock table to be consumed remotely, means that for ENSA2, ASCS can restart in its current location (assuming the node is still available). This change impacts stickiness, migration, and priority parameters. Ensure to use the right command for your enqueue version.

Example

ENSA1

Use the following command to create an **ASCS** `SAPInstance` resource.

```
pcs resource create <rsc_sap_RHX_ASCS00> ocf:heartbeat:SAPInstance
 InstanceName="<RHX_ASCS00_rhxascs>" START_PROFILE=</usr/sap/RHX/SYS/profile/
RHX_ASCS00_rhxascs> AUTOMATIC_RECOVER=false meta resource-stickiness=5000 migration-
threshold=1 op monitor interval=20s on-fail=restart timeout=60s op start interval=0s
 timeout=240s op stop interval=0s timeout=600s --group <grp_RHX_ASCS00>
```

Add a resource stickiness to the group to ensure that ASCS will stay on node, if possible.

```
pcs resource meta <grp_RHX_ASCS00> resource-stickiness=3000
```

Use the following command to create an **ERS** `SAPInstance` resource.

```
pcs resource create <rsc_sap_RHX_ERS10> ocf:heartbeat:SAPInstance
 InstanceName="<RHX_ERS10_rhxers>" START_PROFILE=</usr/sap/RHX/SYS/profile/
RHX_ERS10_rhxers> AUTOMATIC_RECOVER=false IS_ERS=true op monitor interval=20s on-
fail=restart timeout=60s op start interval=0s timeout=600s op stop interval=0s
 timeout=240s --group <grp_RHX_ERS10>
```

ENSA2

Use the following command to create an **ASCS** `SAPInstance` resource.

```
pcs resource create <rsc_sap_RHX_ASCS0> ocf:heartbeat:SAPInstance
 InstanceName="<RHX_ASCS00_rhxascs>" START_PROFILE=</usr/sap/RHX/SYS/profile/
RHX_ASCS00_rhxasc> AUTOMATIC_RECOVER=false meta resource-stickiness=5000 op monitor
 interval=11s on-fail=restart timeout=60s op start interval=0s timeout=600s op stop
 interval=0s timeout=240s --group <grp_RHX_ASCS00>
```

Add a resource stickiness to the group to ensure that ASCS will stay on node, if possible.

```
pcs resource meta <grp_RHX_ASCS00> resource-stickiness=3000
```

Note

`meta resource-stickiness=5000` is here to balance the failover constraint with ERS, so the resource stays on the node where it started, and doesn't migrate around cluster uncontrollably.

Use the following command to create an **ERS** `SAPInstance` resource.

```
pcs resource create <rsc_sap_RHX_ERS10> ocf:heartbeat:SAPInstance
 InstanceName="<RHX_ERS10_rhxers>" START_PROFILE=</usr/sap/RHX/SYS/profile/
RHX_ERS10_rhxers> AUTOMATIC_RECOVER=false op monitor interval=20s on-fail=restart
 timeout=60s op start interval=0s timeout=600s op stop interval=0s timeout=240s --
group <grp_RHX_ERS10>
```

The difference between ENSA1 and ENSA2 is that ENSA2 allows the lock table to be consumed remotely, which means that for ENSA2, ASCS can restart in its current location (assuming the node is still available). This change impacts stickiness, migration and priority parameters. Ensure that you use the right command for your enqueue version.

Create resource constraints

Resource constraints are used to determine where resources run, or alternatively where resources are located, per the conditions. Constraints for SAP NetWeaver ensure that ASCS and ERS are started on separate nodes, and locks are preserved in case of failures. The following are the different types of constraints.

Colocation constraint

The negative score ensures that ASCS and ERS are run on separate nodes, wherever possible.

```
pcs constraint colocation add <grp_RHX_ERS10> with <grp_RHX_ASCS00> -5000
```

Order constraint

This constraint ensures the ASCS instance is started prior to stopping the ERS instance. This can be necessary to consume the lock table.

```
pcs constraint order start <grp_RHX_ASCS00> then stop <grp_RHX_ERS10> symmetrical=false
 kind=Optional
```

Location constraint (ENSA1 only)

This constraint is only required for ENSA1. The lock table can be retrieved remotely for ENSA2, and as a result ASCS doesn’t failover to where ERS is running.

```
pcs constraint location <rsc_sap_RHX_ASCS00> rule score=2000 <runs_ers_RHX> eq 1
```

Activate cluster

Use `pcs config show` to review that all the values have been entered correctly.

On confirmation of correct values, set the maintenance mode to false using the following command. This enables the cluster to take control of the resources.

```
pcs property set maintenance-mode=false
```

See the Sample configuration.

Sample configuration (`pcs config show`)

The following sample configuration is based on ENSA2.

```
[root@rhxhost01 cluster] pcs config show
Cluster Name: rhelha
Corosync Nodes:
 rhxhost01 rhxhost02
Pacemaker Nodes:
 rhxhost01 rhxhost02

Resources:
 Group: grp_RHX_ASCS00
  Meta Attrs: resource-stickiness=3000
  Resource: rsc_fs_RHX_ASCS00 (class=ocf provider=heartbeat type=Filesystem)
   Attributes: device=fs-0f935147fc20d5ce9.efs.ap-south-1.amazonaws.com:/
RHX_ASCS00 directory=/usr/sap/RHX/ASCS00 force_unmount=safe fstype=nfs4
 options=nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
   Operations: monitor interval=20s timeout=40s (rsc_fs_RHX_ASCS00-monitor-
interval-20s)
               notify interval=0s timeout=60s (rsc_fs_RHX_ASCS00-notify-interval-0s)
               start interval=0s timeout=60s (rsc_fs_RHX_ASCS00-start-interval-0s)
               stop interval=0s timeout=60s (rsc_fs_RHX_ASCS00-stop-interval-0s)
  Resource: rsc_ip_RHX_ASCS00 (class=ocf provider=heartbeat type=aws-vpc-move-ip)
   Attributes: interface=eth0 ip=172.16.30.5
 routing_table=rtb-0cbb0e6fa45ac5a93,rtb-06b959bbca4d4df44
   Operations: monitor interval=20s timeout=40s (rsc_ip_RHX_ASCS00-monitor-
interval-20s)
               start interval=0s timeout=180s (rsc_ip_RHX_ASCS00-start-interval-0s)
               stop interval=0s timeout=180s (rsc_ip_RHX_ASCS00-stop-interval-0s)
  Resource: rsc_sap_RHX_ASCS00 (class=ocf provider=heartbeat type=SAPInstance)
   Attributes: AUTOMATIC_RECOVER=false InstanceName=RHX_ASCS00_rhxascs START_PROFILE=/
usr/sap/RHX/SYS/profile/RHX_ASCS00_rhxascs
   Meta Attrs: resource-stickiness=5000
   Operations: demote interval=0s timeout=320s (rsc_sap_RHX_ASCS00-demote-interval-0s)
               methods interval=0s timeout=5s (rsc_sap_RHX_ASCS00-methods-interval-0s)
               monitor interval=11s on-fail=restart timeout=60s (rsc_sap_RHX_ASCS00-
monitor-interval-11s)
               promote interval=0s timeout=320s (rsc_sap_RHX_ASCS00-promote-
interval-0s)
               reload interval=0s timeout=320s (rsc_sap_RHX_ASCS00-reload-interval-0s)
               start interval=0s timeout=600s (rsc_sap_RHX_ASCS00-start-interval-0s)
               stop interval=0s timeout=240s (rsc_sap_RHX_ASCS00-stop-interval-0s)
 Group: grp_RHX_ERS10
```

```
  Resource: rsc_fs_RHX_ERS10 (class=ocf provider=heartbeat type=Filesystem)
   Attributes: device=fs-0f935147fc20d5ce9.efs.ap-south-1.amazonaws.com:/
RHX_ERS10 directory=/usr/sap/RHX/ERS10 force_unmount=safe fstype=nfs4
 options=nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
   Operations: monitor interval=20s timeout=40s (rsc_fs_RHX_ERS10-monitor-interval-20s)
               notify interval=0s timeout=60s (rsc_fs_RHX_ERS10-notify-interval-0s)
               start interval=0s timeout=60s (rsc_fs_RHX_ERS10-start-interval-0s)
               stop interval=0s timeout=60s (rsc_fs_RHX_ERS10-stop-interval-0s)
  Resource: rsc_ip_RHX_ERS10 (class=ocf provider=heartbeat type=aws-vpc-move-ip)
   Attributes: interface=eth0 ip=172.16.30.6
 routing_table=rtb-0cbb0e6fa45ac5a93,rtb-06b959bbca4d4df44
   Operations: monitor interval=20s timeout=40s (rsc_ip_RHX_ERS10-monitor-interval-20s)
               start interval=0s timeout=180s (rsc_ip_RHX_ERS10-start-interval-0s)
               stop interval=0s timeout=180s (rsc_ip_RHX_ERS10-stop-interval-0s)
  Resource: rsc_sap_RHX_ERS10 (class=ocf provider=heartbeat type=SAPInstance)
   Attributes: AUTOMATIC_RECOVER=false InstanceName=RHX_ERS10_rhxers START_PROFILE=/
usr/sap/RHX/SYS/profile/RHX_ERS10_rhxers
   Operations: demote interval=0s timeout=320s (rsc_sap_RHX_ERS10-demote-interval-0s)
               methods interval=0s timeout=5s (rsc_sap_RHX_ERS10-methods-interval-0s)
               monitor interval=20s on-fail=restart timeout=60s (rsc_sap_RHX_ERS10-
monitor-interval-20s)
               promote interval=0s timeout=320s (rsc_sap_RHX_ERS10-promote-interval-0s)
               reload interval=0s timeout=320s (rsc_sap_RHX_ERS10-reload-interval-0s)
               start interval=0s timeout=600s (rsc_sap_RHX_ERS10-start-interval-0s)
               stop interval=0s timeout=240s (rsc_sap_RHX_ERS10-stop-interval-0s)

Stonith Devices:
 Resource: rsc_aws_stonith_RHX (class=stonith type=fence_aws)
  Attributes: pcmk_delay_max=30 pcmk_host_map=rhxhost01:i-
xxxxinstidforhost1;rhxhost02:i-xxxxinstidforhost2 pcmk_reboot_action=reboot
 pcmk_reboot_retries=2 pcmk_reboot_timeout=300 power_timeout=240 region=us-east-1
  Operations: monitor interval=180 timeout=60 (rsc_aws_stonith_RHX-monitor-
interval-180)
              start interval=0s timeout=180 (rsc_aws_stonith_RHX-start-interval-0s)
              stop interval=0s timeout=180 (rsc_aws_stonith_RHX-stop-interval-0s)
Fencing Levels:

Location Constraints:
Ordering Constraints:
  start grp_RHX_ASCS00 then stop grp_RHX_ERS10 (kind:Optional) (non-symmetrical)
 (id:order-grp_RHX_ASCS00-grp_RHX_ERS10-Optional)
Colocation Constraints:
  grp_RHX_ERS10 with grp_RHX_ASCS00 (score:-5000) (id:colocation-grp_RHX_ERS10-
grp_RHX_ASCS00--5000)
```

```
Ticket Constraints:

Alerts:
 No alerts defined

Resources Defaults:
 migration-threshold=3
 resource-stickiness=1
Operations Defaults:
 No defaults set

Cluster Properties:
 cluster-infrastructure: corosync
 cluster-name: rhelha
 dc-version: 1.1.23-1.el7_9.1-9acf116022
 have-watchdog: false
 last-lrm-refresh: 1674393658
 maintenance-mode: false

Quorum:
  Options:
```

Operations

This section covers the following topics.

Topics

- Analysis and maintenance
- Testing

Analysis and maintenance

This section covers the following topics.

Topics

- Viewing the cluster state
- Performing planned maintenance
- Post-failure analysis and reset
- Alerting and monitoring

Viewing the cluster state

You can view the state of the cluster based on your operating system.

Operating system based

There are multiple operating system commands that can be run as root or as a user with appropriate permissions. The commands enable you to get an overview of the status of the cluster and its services. See the following commands for more details.

```
pcs status
```

Sample output:

```
pcs status
Cluster name: rhelha
Stack: corosync
Current DC: <rhxhost01> (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Sat Jan 14 20:55:26 2023
Last change: Sat Jan 14 17:45:40 2023 by hacluster via crmd on <rhxhost01>

2 nodes configured
7 resource instances configured

Online: [ <rhxhost01> <rhxhost02> ]

Full list of resources:

 rsc_aws_stonith_RHX    (stonith:fence_aws):    Started <rhxhost01>
 Resource Group: grp_RHX_ASCS00
     rsc_fs_RHX_ASCS00  (ocf::heartbeat:Filesystem):    Started <rhxhost01>
     rsc_ip_RHX_ASCS00  (ocf::heartbeat:aws-vpc-move-ip):       Started <rhxhost01>
     rsc_sap_RHX_ASCS00 (ocf::heartbeat:SAPInstance):   Started <rhxhost01>
 Resource Group: grp_RHX_ERS10
     rsc_fs_RHX_ERS10   (ocf::heartbeat:Filesystem):    Started <rhxhost02>
     rsc_ip_RHX_ERS10   (ocf::heartbeat:aws-vpc-move-ip):       Started <rhxhost02>
     rsc_sap_RHX_ERS10  (ocf::heartbeat:SAPInstance):   Started <rhxhost02>

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

The following table provides a list of useful commands.

Command	Description
`crm_mon`	Display cluster status on the console with updates as they occur
`crm_mon -1`	Display cluster status on the console just once, and exit
`crm_mon -Arnf`	-A Display node attributes -n Group resources by node -r Display inactive resources -f Display resource fail counts
`pcs help`	View more options
`crm_mon --help-all`	View more options

Performing planned maintenance

The cluster connector is designed to integrate the cluster with SAP start framework (`sapstartsrv`), including the rolling kernel switch (RKS) awareness. Stopping and starting the SAP system using `sapcontrol` should not result in any cluster remediation activities as these actions are not interpreted as failures. Validate this scenario when testing your cluster.

There are different options to perform planned maintenance on nodes, resources, and the cluster.

Topics

- Maintenance mode
- Placing a node in standby mode
- Moving a resource (not recommended)

Maintenance mode

Use maintenance mode if you want to make any changes to the configuration or take control of the resources and nodes in the cluster. In most cases, this is the safest option for administrative tasks.

On

- Use the following command to turn on maintenance mode.

  ```
  pcs property set maintenance-mode="true"
  ```

Off

- Use the following command to turn off maintenance mode.

  ```
  pcs property set maintenance-mode="false"
  ```

Placing a node in standby mode

To perform maintenance on the cluster without system outage, the recommended method for moving active resources is to place the node you want to remove from the cluster in standby mode.

```
pcs node standby <rhxhost01>
```

The cluster will cleanly relocate resources, and you can perform activities, including reboots on the node in standby mode. When maintenance activities are complete, you can re-introduce the node with the following command.

```
pcs node unstandby <rhxhost01>
```

Moving a resource (not recommended)

Moving individual resources is not recommended because of the migration or move constraints that are created to lock the resource in its new location. These can be cleared as described in the info messages, but this introduces an additional setup.

```
<rhxhost01>:~ pcs resource move <grp_RHX_ASCS00 rhxhost02>

Note: Move constraint created for <grp_RHX_ASCS00 to rhxhost02>
```

```
Note: Use “pcs constraint location remove cli-prefer-<grp_RHX_ASCS00>” to remove this
 constraint.
```

Post-failure analysis and reset

A review must be conducted after each failure to understand the source of failure as well the reaction of the cluster. In most scenarios, the cluster prevents an application outage. However, a manual action is often required to reset the cluster to a protective state for any subsequent failures.

Topics

- Checking the logs
- Cleanup pcs status
- Restart failed nodes or pacemaker
- Further analysis

Checking the logs

Start your troubleshooting by checking the operating system log `/var/log/messages`. You can find additional information in the cluster and pacemaker logs.

- **Cluster logs** – updated in the `corosync.conf` file located at `/etc/corosync/corosync.conf`.
- **Pacemaker logs** – updated in the `pacemaker.log` file located at `/var/log/pacemaker`.
- **Resource agents** – `/var/log/messages`

Application based failures can be investigated in the SAP work directory.

Cleanup `pcs status`

If failed actions are reported using the `crm status` command, and if they have already been investigated, then you can clear the reports with the following command.

```
pcs resource cleanup <resource> <hostname>
```

```
pcs stonith cleanup
```

Note

Use the help command to understand the impact of these commands.

Restart failed nodes or `pacemaker`

It is recommended that failed (or fenced) nodes are not automatically restarted. It gives operators a chance to investigate the failure, and ensure that the cluster doesn't make assumptions about the state of resources.

You need to restart the instance or the pacemaker service based on your approach.

Further analysis

If further analysis from Red Hat is required, they may request an sos report, or logs of the cluster from `crm_report` or `pcs cluster report`.

sos report – The sos report command is a tool that collects configuration details, system information, and diagnostic information from a Red Hat Enterprise Linux system. For instance, the running kernel version, loaded modules, and system and service configuration files. The command also runs external programs to collect further information, and stores this output in the resulting archive. For more information, see Red Hat documentation What is an sos report and is it different from an sosreport?

crm report – collects the cluster logs/information from the node where the command is being run. For more information, see Red Hat documentation How do I generate a crm_report from a RHEL 6 or 7 High Availability cluster node using pacemaker?

```
crm_report
```

pcs cluster report – command collects the cluster logs/information from all the nodes involved in the cluster.

```
pcs cluster report <destination_path>
```

Note

The `pcs cluster report` command relies on passwordless ssh being set up between the nodes.

Alerting and monitoring

This section covers the following topics.

Topics

- Using Amazon CloudWatch Application Insights
- Using the cluster alert agents

Using Amazon CloudWatch Application Insights

For monitoring and visibility of cluster state and actions, Application Insights includes metrics for monitoring enqueue replication state, cluster metrics, and SAP and high availability checks. Additional metrics, such as EFS and CPU monitoring can also help with root cause analysis.

For more information, see Get started with Amazon CloudWatch Application Insights and SAP NetWeaver High Availability on Amazon EC2.

Using the cluster alert agents

Within the cluster configuration, you can call an external program (an alert agent) to handle alerts. This is a *push* notification. It passes information about the event via environment variables.

The agents can then be configured to send emails, log to a file, update a monitoring system, etc. For example, the following script can be used to access Amazon SNS.

```
#!/bin/sh

alert_sns.sh
modified from /usr/share/pacemaker/alerts/alert_smtp.sh.sample

##############################################################################
#SETUP
* Create an SNS Topic and subscribe email or chatbot
* Note down the ARN for the SNS topic
* Give the IAM Role attached to both Instances permission to publish to the SNS Topic
* Ensure the aws cli is installed
* Copy this file to /usr/share/pacemaker/alerts/alert_sns.sh or other location on BOTH
 nodes
* Ensure the permissions allow for hacluster and root to execute the script
* Run the following as root (modify file location if necessary and replace SNS ARN):

#SLES:
    crm configure alert aws_sns_alert /usr/share/pacemaker/alerts/alert_sns.sh meta
 timeout=30s timestamp-format="%Y-%m-%d_%H:%M:%S" to <{ arn:aws:sns:region:account-
id:myPacemakerAlerts  }>
#RHEL:
    pcs alert create id=aws_sns_alert path=/usr/share/pacemaker/alerts/alert_sns.sh
 meta timeout=30s timestamp-format="%Y-%m-%d_%H:%M:%S"
    pcs alert recipient add aws_sns_alert value=<arn:aws:sns:region:account-
id:myPacemakerAlerts>

#Additional information to send with the alerts.
node_name=`uname -n`
sns_body=`env | grep CRM_alert_`

#Required for SNS
TOKEN=$(/usr/bin/curl --noproxy '*' -s -X PUT "http://169.254.169.254/latest/api/token"
 -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

#Get metadata
REGION=$(/usr/bin/curl --noproxy '*' -w "\n" -s -H "X-aws-ec2-metadata-token: $TOKEN"
 http://169.254.169.254/latest/dynamic/instance-identity/document | grep region | awk -
F\" '{print $4}')

sns_subscription_arn=${CRM_alert_recipient}

#Format depending on alert type
case ${CRM_alert_kind} in
   node)
     sns_subject="${CRM_alert_timestamp} ${cluster_name}: Node '${CRM_alert_node}' is
 now '${CRM_alert_desc}'"
   ;;
   fencing)
     sns_subject="${CRM_alert_timestamp} ${cluster_name}: Fencing ${CRM_alert_desc}"
   ;;
   resource)
     if [ ${CRM_alert_interval} = "0" ]; then
         CRM_alert_interval=""
     else
```

```
         CRM_alert_interval=" (${CRM_alert_interval})"
     fi
     if [ ${CRM_alert_target_rc} = "0" ]; then
         CRM_alert_target_rc=""
     else
         CRM_alert_target_rc=" (target: ${CRM_alert_target_rc})"
     fi
     case ${CRM_alert_desc} in
         Cancelled)
           ;;
         *)
           sns_subject="${CRM_alert_timestamp}: Resource operation
 '${CRM_alert_task}${CRM_alert_interval}' for '${CRM_alert_rsc}' on
 '${CRM_alert_node}': ${CRM_alert_desc}${CRM_alert_target_rc}"
           ;;
     esac
     ;;
   attribute)
     sns_subject="${CRM_alert_timestamp}: The '${CRM_alert_attribute_name}' attribute
 of the '${CRM_alert_node}' node was updated in '${CRM_alert_attribute_value}'"
     ;;
   *)
     sns_subject="${CRM_alert_timestamp}: Unhandled $CRM_alert_kind alert"
     ;;
esac

#Use this information to send the email.
aws sns publish --topic-arn "${sns_subscription_arn}" --subject "${sns_subject}" --
message "${sns_body}" --region ${REGION}
```

Testing

We recommend scheduling regular fault scenario recovery testing at least annually, and as part of the operating system or SAP kernel updates that may impact operations. For more details on best practices for regular testing, see SAP Lens – Best Practice 4.3 – Regularly test business continuity plans and fault recovery.

The tests described here simulate failures. These can help you understand the behavior and operational requirements of your cluster.

In addition to checking the state of cluster resources, ensure that the service you are trying to protect is in the required state. Can you still connect to SAP? Are locks still available in SM12?

Define the recovery time to ensure that it aligns with your business objectives. Record recovery actions in runbooks.

Topics

Test 1: Stop ASCS on the primary node using `sapcontrol`

Notes – Ensure that the connector has been installed and the parameters have been updated.

Simulate failure – On `<rhxhost01>` as `rhxadm`:

```
sapcontrol -nr <00> -function Stop
```

Expected behavior – ASCS should be stopped on `<rhxhost01>`, and the cluster should not perform any activity.

Recovery action – Start ASCS manually.

Test 2: Stop ERS on the secondary node using `sapcontrol`

Notes – Ensure that the connector has been installed, and the parameters are updated.

Simulate failure – On `<rhxhost02>` as `rhxadm`:

```
sapcontrol -nr <10> -function Stop
```

Expected behavior – ERS should be stopped on `<rhxhost02>`, and the cluster should not perform any activity.

Recovery action – Start ERS manually.

Test 3: Kill the message server process on the primary node

Simulate failure – On `<rhxhost01>` as root:

```
kill -9 $(pgrep -f "ms.sap<RHX>_ASCS<00>")
```

Expected behavior – The message server should immediately respawn based on the Restart parameter.

Recovery action – No action required.

Test 4: Kill the enqueue server process on the primary node

Notes – Check that locks have persisted, and review the location constraints that only exist for ENSA1.

Simulate failure – On `<rhxhost01>` as `rhxadm`:

```
kill -9 $(pgrep -f "[en|enq].sap<RHX>_ASCS<00>")
```

Expected behavior – ENSA2: Cluster will restart the ENQ process and retrieve the locks remotely. ENSA1: Cluster will failover the ASCS resource to the node where the ERS is running.

Recovery action – No action required.

Test 5: Kill the ER process

Simulate failure – On `<rhxhost02>` as root:

```
kill -9 $(pgrep -f "[er|enqr].sap<RHX>_ERS<10>")
```

Expected behavior – Cluster will restart the ERS on the same node.

Recovery action – No action required.

Test 6: Simulate hardware failure of an individual node, and repeat for other node

Notes – To simulate a system crash, you must first ensure that `/proc/sys/kernel/sysrq` is set to 1.

Simulate failure – On both nodes as root:

```
echo 'b' > /proc/sysrq-trigger
```

Expected behavior – The node which has been killed fails. The cluster will move the resources (ASCS/ERS) which were running on the failed node to the surviving node.

Recovery action – Start the EC2 node and pacemaker service. The cluster will detect that the node is online and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Test 7: Simulate a network failure

Notes – See the following list.

- Iptables must be installed.
- Check for any existing iptables rules as iptables -F will flush all rules.
- Review pcmk_delay and priority parameters if you see neither node survives the fence race.

Simulate failure – On either node as root:

```
iptables -A INPUT -s <CIDR_of_other_subnet> -j DROP; iptables -A OUTPUT -d
 <CIDR_of_other_subnet> -j DROP
```

Expected behavior – The cluster detects the network failure, and fences one of the nodes to avoid a split-brain situation.

Recovery action – If the node where the command was run survives, execute iptables -F to clear the network failure. Start the EC2 node and pacemaker service. The cluster will detect that the node is online and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Test 8: Simulate an NFS failure

Notes – See the following list.

- Iptables must be installed.
- Check for any existing iptables rules as iptables -F will flush all rules.

- Although rare, this is an important scenario to test. Depending on the activity it may take some time (10 min +) to notice that I/O to EFS is not occurring and fail either the Filesystem or SAP resources.

Simulate failure – On either node as root:

```
iptables -A OUTPUT -p tcp --dport 2049 -m state --state NEW,ESTABLISHED,RELATED -j
 DROP; iptables -A INPUT -p tcp --sport 2049 -m state --state ESTABLISHED -j DROP
```

Expected behavior – The cluster detects that NFS is not available, and the SAP Instance resource agent will fail and move to the FAILED state. Because of the option "on-fail=restart" configuration, the cluster will try a local restart before eventually fencing the node and failing over.

Recovery action – If the node where the command was run survives, execute iptables -F to clear the network failure. Start the EC2 node and pacemaker service. The cluster will detect that the node is online and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Test 9: Accidental shutdown

Notes – See the following list.

- Avoid shutdowns without cluster awareness.
- We recommend the use of systemd to ensure predictable behaviour.
- Ensure the resource dependencies are in place.

Simulate failure – Login to AWS Management Console, and stop the instance or issue a shutdown command.

Expected behavior – The node which has been shut down fails. The cluster will move the resources (ASCS/ERS) which were running on the failed node to the surviving node. If systemd and resource dependencies are not configured, you may notice that while the EC2 instance is shutting down gracefully, the cluster will detect an unclean stop of cluster services on the node and will fence the EC2 instance being shut down.

Recovery action – Start the EC2 node and pacemaker service. The cluster will detect that the node is online, and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Migrate SAP NetWeaver applications with AWS Migration Hub Orchestrator

AWS Migration Hub Orchestrator simplifies and automates the migration of servers and enterprise applications to AWS. It provides a single location to run and track your migrations. It helps reduce migration costs and time by automating many migration tasks. Migration Hub Orchestrator offers templates to create a migration workflow that can be customized to fit your unique migration requirements.

With Migration Hub Orchestrator, you can migrate SAP NetWeaver based applications running on SAP HANA or any other database, such as Oracle, MSSQL, SAP ASE, etc., to AWS. For more information, see What is AWS Migration Hub Orchestrator?

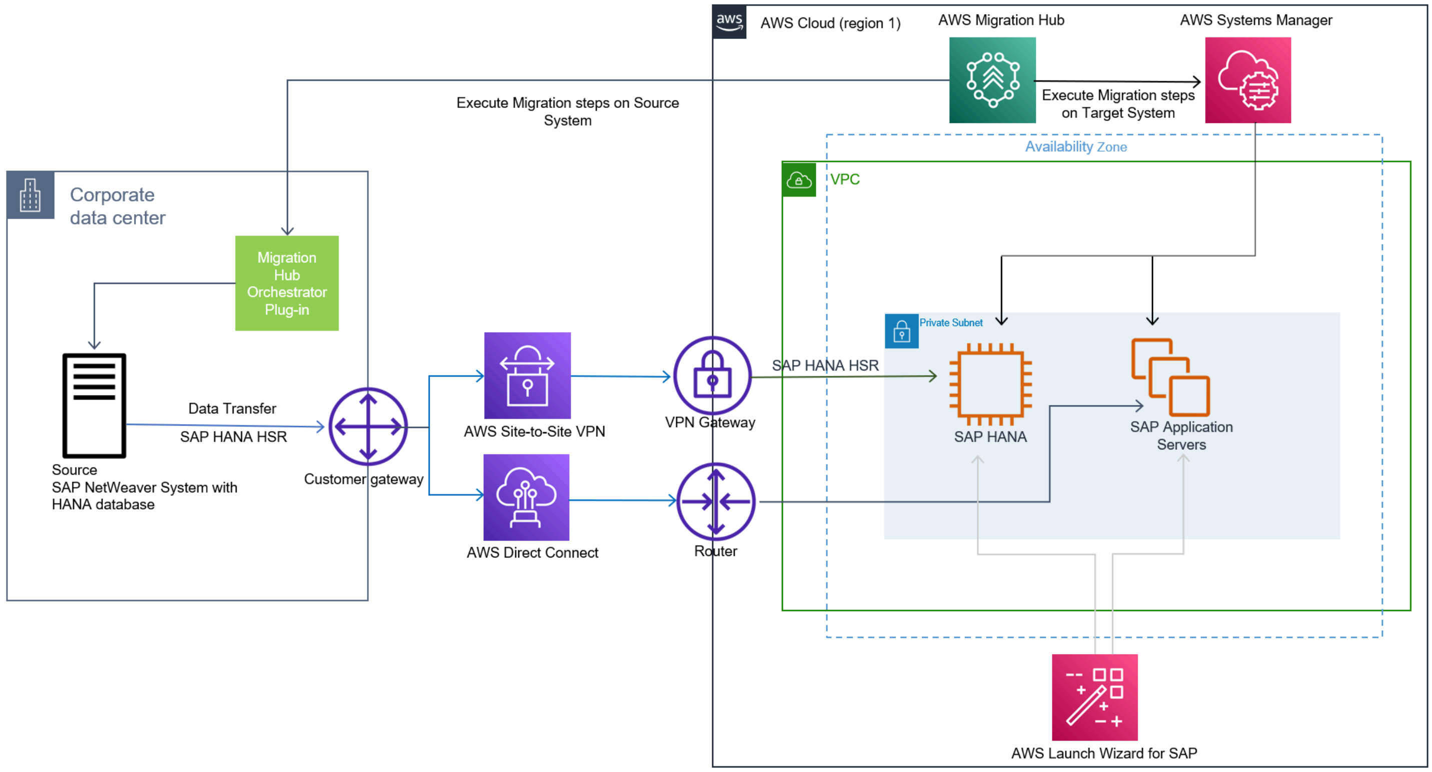
You can access AWS Migration Hub Orchestrator from link: https://console.aws.amazon.com/migrationhub/orchestrator/ or from the AWS Command Line Interface.

Topics

Migrate applications with SAP HANA

To migrate SAP NetWeaver based applications running on SAP HANA database, use the Migrate SAP NetWeaver based applications and SAP HANA databases to AWS template.

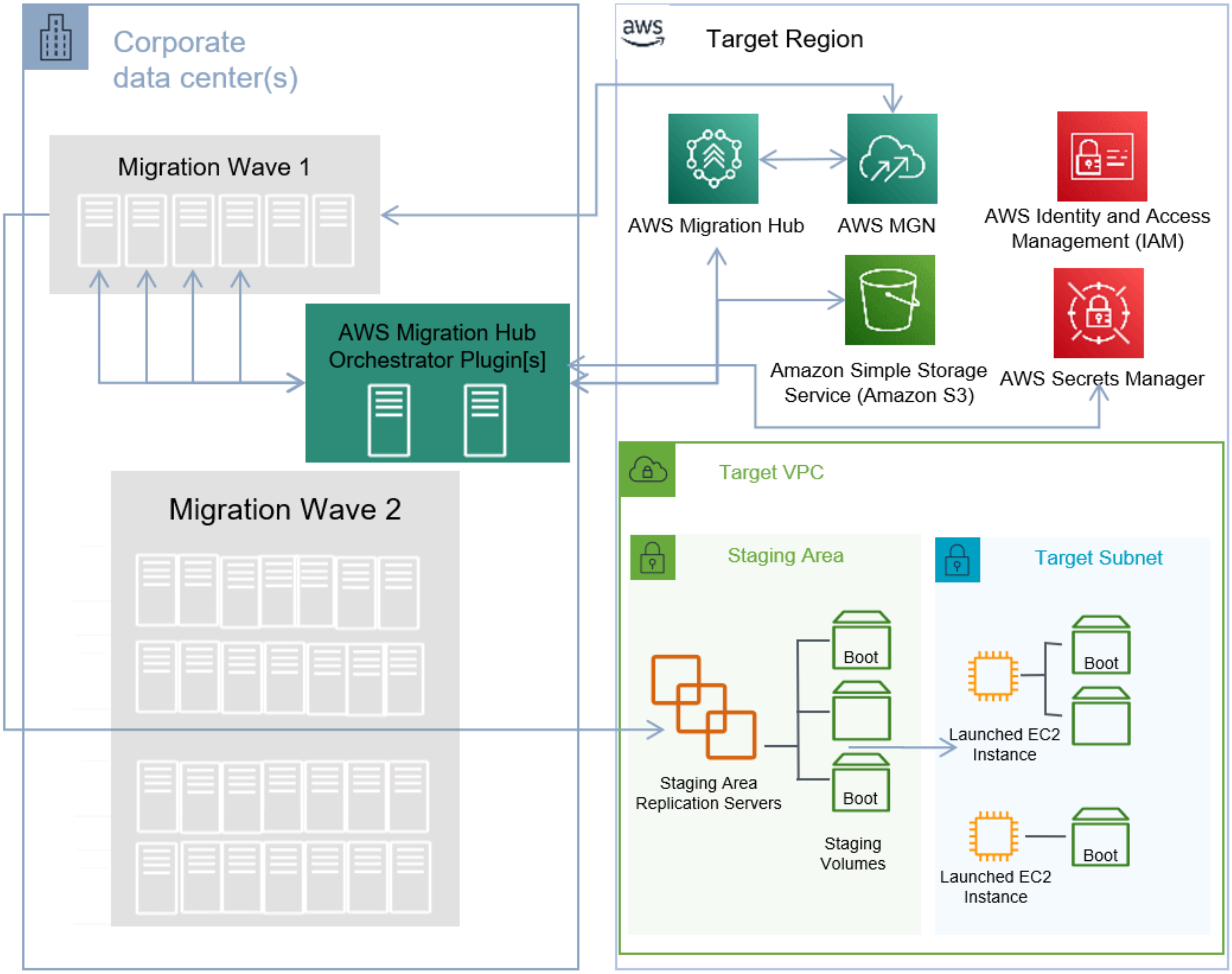
The following diagram illustrates an application migration with this template.



Migrate applications with any database

To migrate SAP NetWeaver based applications running on any database *other than SAP HANA*, use the Rehost applications on Amazon EC2 Migration Hub Orchestrator template.

The following diagram illustrates an application migration with this template.


Corporate data center(s)
Migration Wave 1
AWS Migration Hub Orchestrator Plugin[s]
Migration Wave 2
Target Region
AWS Migration Hub
AWS MGN
AWS Identity and Access Management (IAM)
Amazon Simple Storage Service (Amazon S3)
AWS Secrets Manager
Target VPC
Staging Area
Target Subnet
Boot
Boot
Staging Area Replication Servers
Staging Volumes
Boot
Launched EC2 Instance
Boot
Launched EC2 Instance

Oracle for SAP NetWeaver on AWS Deployment and Operations Guide for Linux

SAP specialists, Amazon Web Services

December 2021

This guide is part of a content series that provides detailed information about hosting, configuring, and using SAP technologies in the Amazon Web Services Cloud. For more information, see SAP on AWS Technical Documentation.

Overview

The purpose of this guide is to provide an overview of how to implement and operate Oracle database for SAP NetWeaver applications on Amazon Elastic Compute Cloud (Amazon EC2). It is for users who are responsible for planning, architecting, and deploying Oracle database for SAP NetWeaver applications on AWS. You should have a good understanding of AWS services, general network concepts, Oracle Enterprise Linux (OEL) OS, and Oracle database administration. This guide provides guidance to successfully launch and configure the resources required for Oracle database on AWS.

It doesn’t provide guidance on how to setup network and security components like Amazon Virtual Private Cloud (Amazon VPC), subnets, route tables, ACLs, NAT gateway, AWS Identity and Access Management (IAM) roles, and AWS security groups. It focuses on how to configure and maintain the compute, storage, and OS components for the Oracle database on Linux on AWS. It is not intended to replace the standard installation and administration guides from SAP or Oracle.

Prerequisites

We recommend familiarizing yourself with these guides:

- SAP on AWS Overview and Planning
- Getting Started with Architecting SAP on the AWS Cloud
- Best practices for Amazon EC2
- Migrating Oracle Database Workloads to Oracle Linux on AWS
- Determining the IOPS Needs for Oracle Database on AWS

- SAP Note 2606828 - Oracle Database Roadmap for SAP NetWeaver (SAP portal access required)

Technical requirements

Before you begin deploying Oracle database for SAP applications on AWS, ensure that you meet the following requirements:

- If necessary, request a service limit increase by creating a support ticket. This is to ensure that the AWS services required for Oracle database deployment are not constrained by the default limit. For more information, see AWS service quotas. For example, you may have to increase the Amazon EC2 instance limit before your Oracle deployment.
- You will need the following information for your existing resources while running the AWS CLI commands to create Amazon EC2 and Amazon Elastic Block Store (Amazon EBS) resources.

Information

Information	Description
AWS Region	Region where you want to deploy your AWS resources.
Availability Zone (AZ)	Availability Zone within your target Region where you want to deploy your resources.
Amazon VPC id	Amazon VPC where you want to deploy your Amazon EC2 instances for SAP installation.
VPS subnet id	Subnet where you want to deploy your Amazon EC2 instances.
Linux AMI id	Amazon Machine Image (AMI) that will be used to launch your Amazon EC2 instances . You can find the latest Linux AMIs on AWS Marketplace.
Key pair	Make sure that you have generated the key pair in your target Region and that you have access to the private key.

Security group id	Name of the security group that you want to assign to your Amazon EC2 instances.
Access key ID	Access key for your AWS account that will be used with AWS CLI tools.
Secret access key	Secret key for your AWS account that will be used with AWS CLI tools.

- Create security groups and open ports to enable communication. For existing security groups, ensure that the required ports are open. For a list of ports, refer to TCP/IP ports of all SAP products and Managing Oracle Database Port Numbers.
- Ensure that you have installed and configured AWS CLI with required credentials, if you plan to use it to launch instances. For more information, see Installing the AWS CLI.
- If you plan to use the AWS Management Console, ensure that you have the essential credentials and permissions to launch and configure AWS services. For more information, see Access management for AWS resources.
- Ensure that you have the software files required for installation readily available. You can stage these in Amazon S3 or Amazon Elastic File System (Amazon EFS). Amazon EFS can be easily shared on all of your installation hosts. For more information, see Create your Amazon EFS file system.
- Oracle for SAP on AWS is supported on an OEL OS. For more information, see SAP Note 1656099 and SAP Note 2358420 (login required). If you are currently using a different OS, you can procure licenses and perform a migration. For more information, see Migrating Oracle Database Workloads. To use AMIs published by Oracle, see Launch an Oracle Linux instance in AWS.

Planning

Plan your SAP system landscape according to the SAP Master Guide for your version of SAP NetWeaver for Linux OS and Oracle database.

- Deployment options
- Sizing
- Amazon Machine Image (AMI)
- Security and compliance

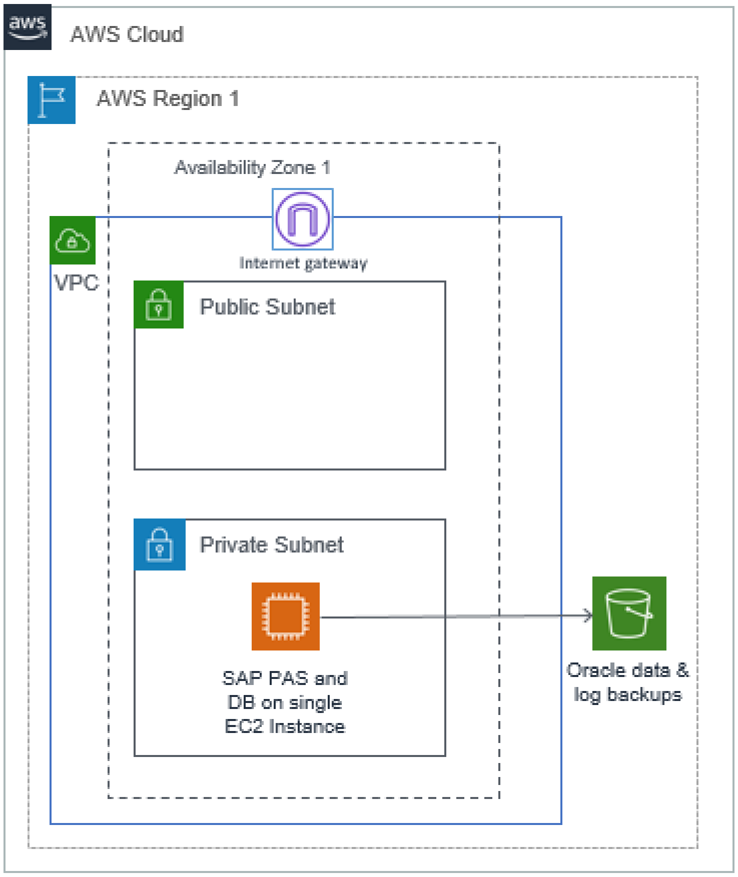
- Storage for Oracle

Deployment options

To install Oracle for SAP NetWeaver, you have four deployment options:

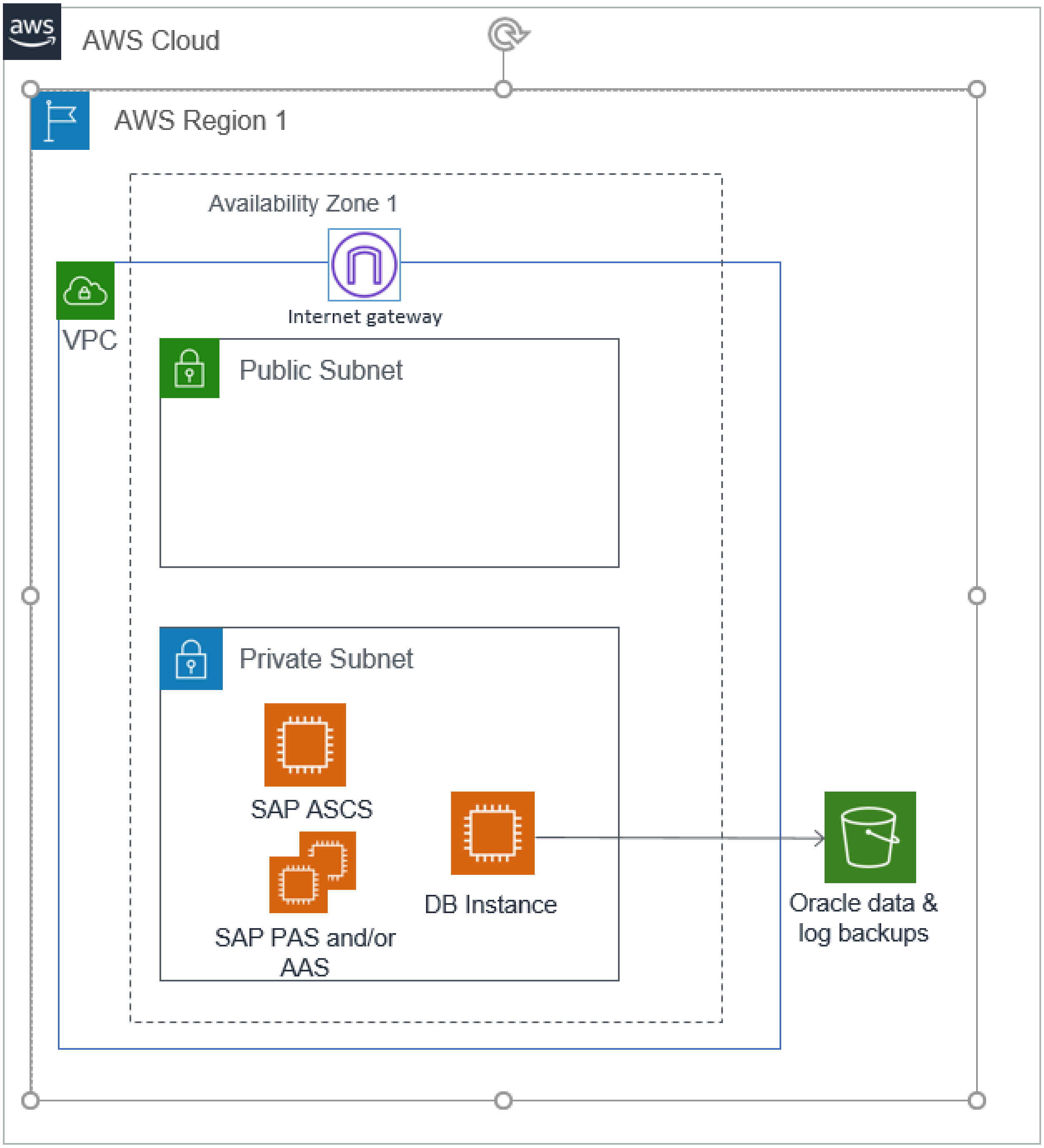
Standalone deployment

In standalone deployment (also known as single host installation), all components of the SAP NetWevaer, ABAP SAP Central Services (ASCS), and database Primary Application Server (PAS) run on one Amazon EC2 instance. One Amazon EC2 instance in a single Availability Zone in a single Region runs the Oracle database. This option can be optimal for non-production workloads. You can use Amazon EC2 auto recovery feature to protect your instance against infrastructure issues like loss of network connectivity or system power. However, this solution is not database state aware and does not protect your database against storage failure, OS issues, Availability Zone or Region failure.



Distributed deployment

In distributed deployment, every instance of SAP NetWeaver (ASCS/SCS, database, PAS, and optionally AAS) can run on a separate Amazon EC2 instance. This system also deploys Oracle database in a single Availability Zone. You can use Amazon EC2 auto recovery feature to protect your instance against infrastructure issues like loss of network connectivity or system power.



High availability deployment

In high availability deployment, you deploy two Amazon EC2 instances across two Availability Zones within a Region, and the Oracle database with Oracle Data Guard or a third-party high availability solution.

> **Note**
>
> When using native Oracle with SAP and AWS features, the design must be a subset of supported features, as described in SAP Note 105047 and SAP Note 2358420.

Option 1: high availability with Oracle Data Guard

Oracle Data Guard is a feature of the Oracle database enterprise edition. It provides a set of tools to manage one or more Oracle standby databases for high availability and disaster recovery. To create an Oracle standby database, replicate the Oracle primary database to a secondary server by backup/restore or RMAN duplicate method. When the standby database is set up, any changes to the primary database are replicated on the standby database. This ensures that both the databases are in sync. The following table describes the replication methods associated with the Oracle Data Guard protection modes.

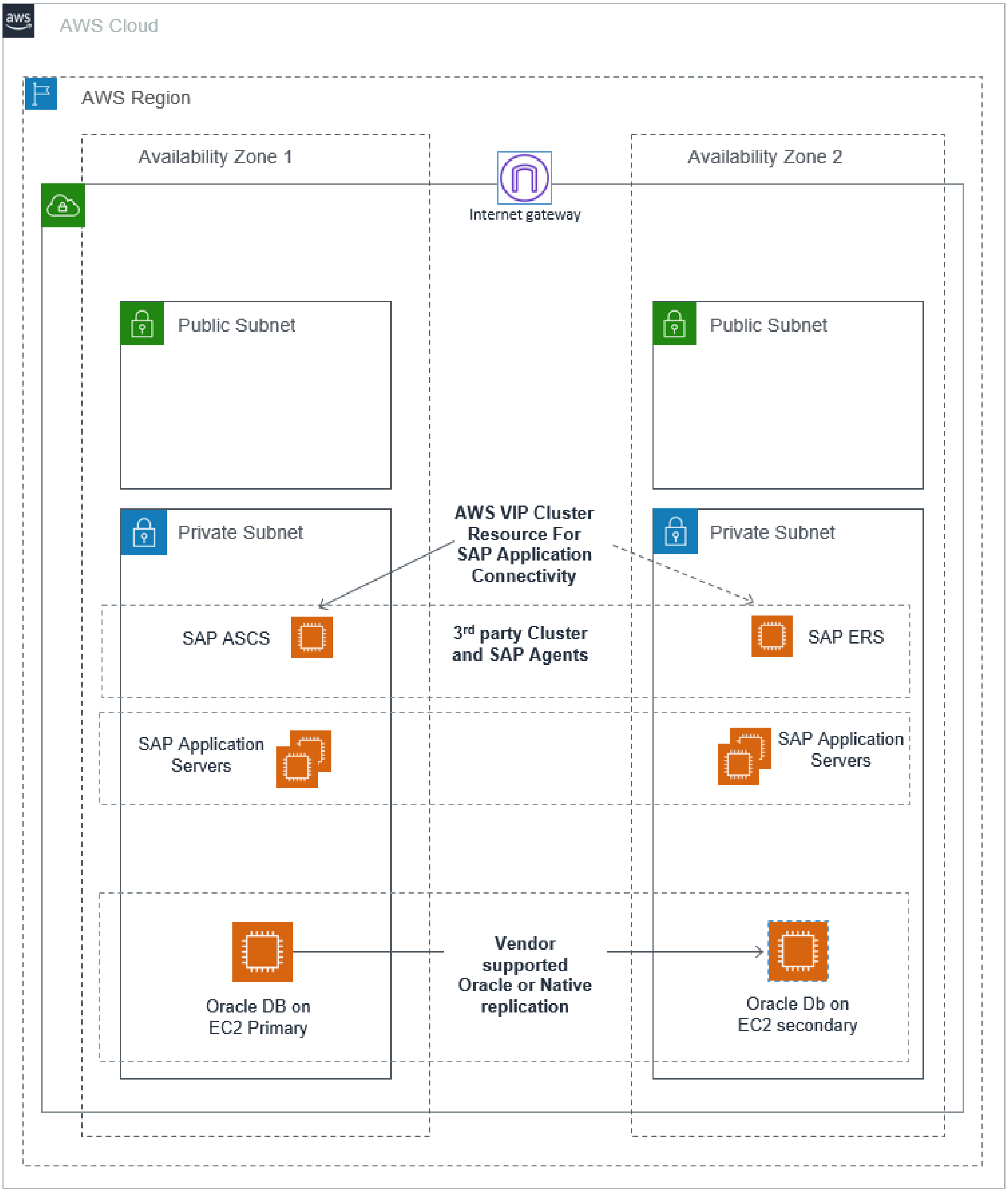
Oracle Data Guard protection modes

Protection mode	Replication	Behavior
Maximum performance	Asynchronous	Primary database is not affected by any delays in writing redo data to the standby database.
Maximum availability	Synchronous	Commit occurs when all the redo data needed to recover transactions has been written to the online redo log and to at least one synchronized standby database. If Data Guard is not able to write to the standby database, the

		behavior will be similar to the maximum performance protection mode.
Maximum protection	Synchronous	Changes must be written to both, the online redo log and to the standby database for every transaction. If Data Guard is unable to write the redo stream to at least one standby, it will shut down the primary database.

All Availability Zones within an AWS Region are connected with high-bandwidth over fully redundant and dedicated metro fiber, providing high-throughput and low-latency networking between Availability Zones. For a high availability configuration, you can set up a primary and standby relationship between two Oracle databases with synchronous replication, running on Amazon EC2 instances in different Availability Zones within the same Region.

The maximum protection mode can cause a shutdown of the primary database in case of a standby database failure. Unless you need to meet a compliance requirement, we recommend using the maximum availability option for high availability.


AWS Cloud
AWS Region
Availability Zone 1
Availability Zone 2
Internet gateway
Public Subnet
Public Subnet
Private Subnet
Private Subnet
AWS VIP Cluster Resource For SAP Application Connectivity
SAP ASCS
3rd party Cluster and SAP Agents
SAP ERS
SAP Application Servers
SAP Application Servers
Oracle DB on EC2 Primary
Vendor supported Oracle or Native replication
Oracle Db on EC2 secondary

You can use manual failover or switchover to the standby database by following the steps in the Data Guard Broker Switchover and Failover Operations. Alternatively, you can automate this process. For more information, see Oracle Data Guard Fast-Start Failover. To reconnect the SAP applications after the failover is complete, refer to the Reconnect SAP instance to database section in the https://www.sap.com/documents/2016/12/a67bac51-9a7c-0010-82c7-eda71af511fa.html.

Option 2: high availability using third-party products

You can use third-party products to achieve Oracle database high availability in your SAP on AWS environment. Here are two examples:

- Using SIOS to Protect your Critical Core on AWS
- High Availability Configuration in AWS Cloud using InfoScale Enterprise

For a complete list of certified partners, see the SAP wiki.

These products provide end-to-end high availability for SAP applications and databases. They also detect failures and perform automatic failovers, making them a good option for production environments with low recovery time objective. Using a virtual IP address makes the user or application redirection automatic in case of failover. For more information, see the vendor documentation.

Both of the preceding mentioned third-party examples are using their own storage (Data Keeper for SIOS and Veritas Volume Replicator for Veritas) and not database native replication. In option 1, the database is replicated using the Oracle Data Guard. The Guard supports SIOS but is not controlled by the SIOS application recovery kits, that is, a database failover is handled by the Guard. The Guard also supports Veritas, you can replicate using either the Guard or the Veritas Volume Replicator.

Disaster recovery deployment

With disaster recovery deployment of your SAP systems on AWS Cloud, you can achieve business continuity. Based on recovery time objective, recovery point objective, and cost, you can set up disaster recovery deployment with one of the following three scenarios:

- backup and restore
- pilot light
- hot standby

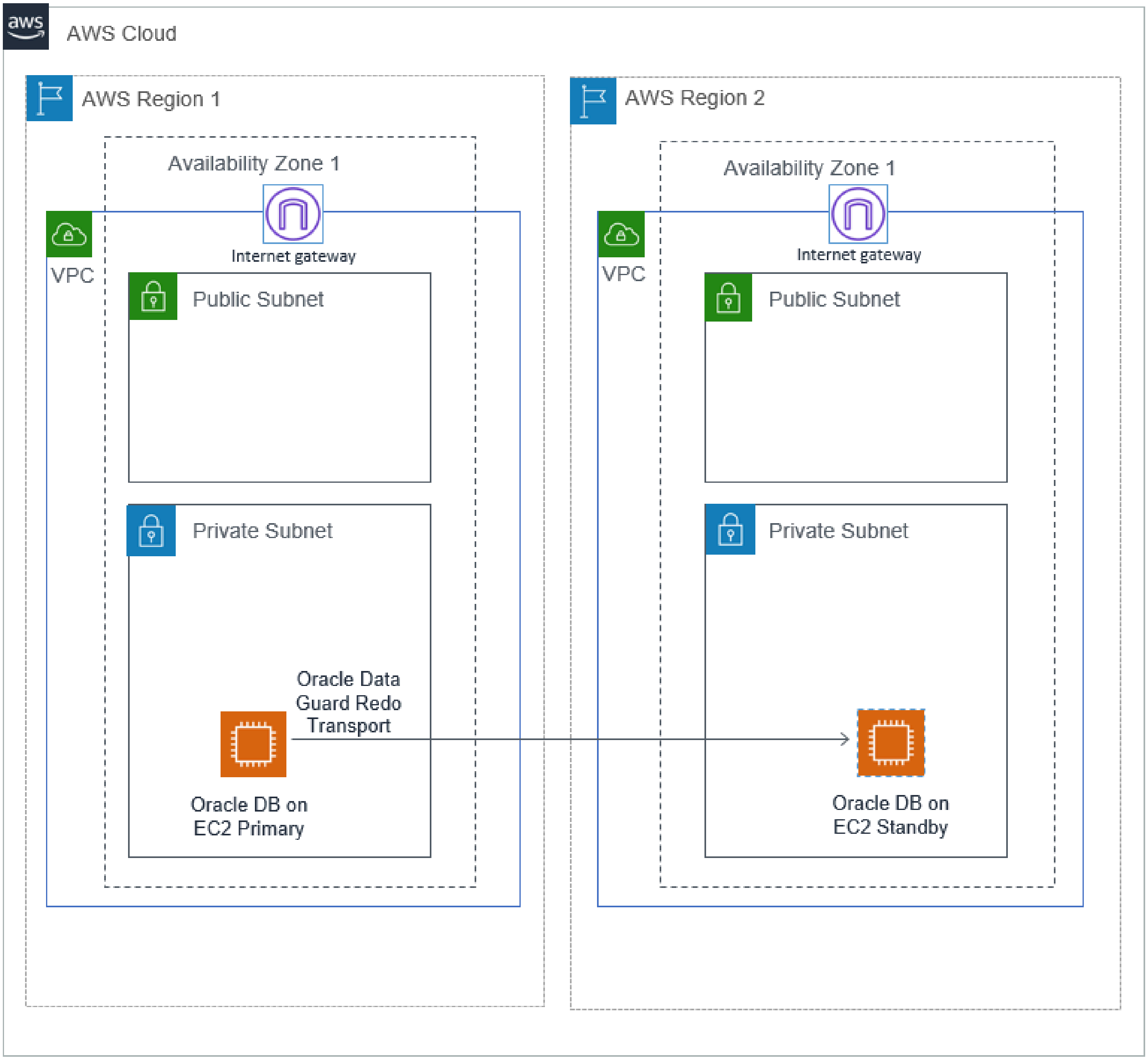
To setup disaster recovery across AWS Regions, setup additional Amazon EC2 instances in a secondary Region for standby Oracle database. Also, configure Oracle Data Guard or a third-part solution for data replication across AWS Regions.

Option 1:disaster recovery using Oracle Data Guard

Using Oracle Data Guard, you can set up pilot light or hot standby DR deployment in your AWS Region. The maximum performance (asynchronous copy) option must be used in Data Guard.

Hot standby Amazon EC2 instance is of the same size as your production Amazon EC2 instance. This makes it more efficient to switch over during a DR test or event. Alternatively, you can use a pilot light approach wherein a smaller size Amazon EC2 instance is running in the AWS DR Region as the target of data replication. This Amazon EC2 instance should have enough resources to take over the load of data replication. This option costs less than hot standby. However, during a DR test or event, you have to perform an additional step of resizing the Amazon EC2 instance. Before switchover, you must resize the DR Amazon EC2 instance to the same size as production, so that it can take over the production workloads. This step increases the time required to switchover to DR. You must also factor in the latency of running a non-production environment in another Region.

Pilot light option can run a non-production environment that is the same size as production in the DR Region. This ensures availability of Amazon EC2 instance in case of a DR event.

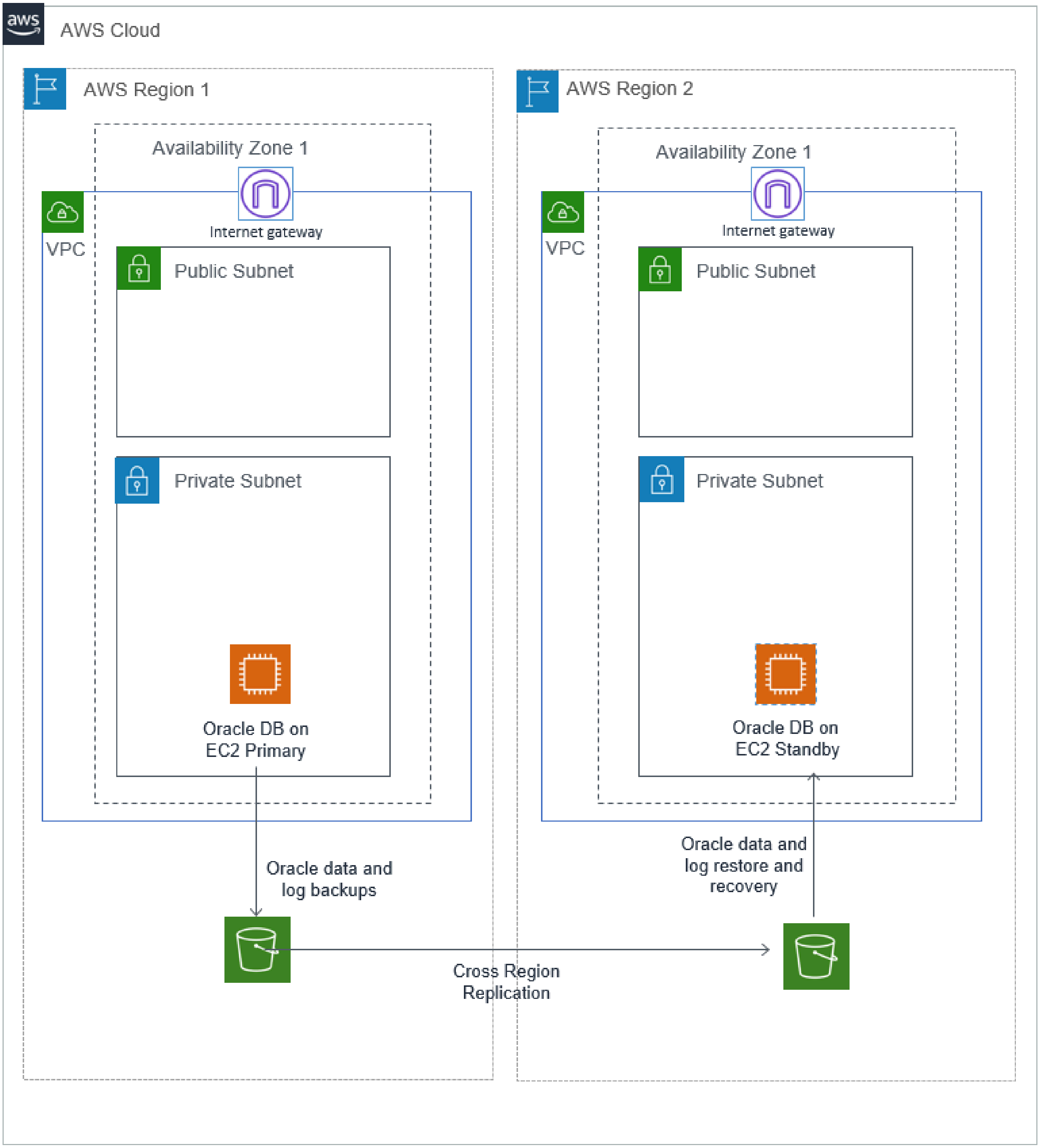


Option 2:passive disaster recovery using backup and recovery

This option uses Oracle database backup and recovery feature to set up your DR. You can store your database backups in Amazon Simple Storage Service (Amazon S3) and use the Amazon S3 Cross-region replication (CRR) to replicate your backup to your target Region. It enables automatic, asynchronous copying of objects across buckets in different AWS Regions. You can install and configure Oracle database on an Amazon EC2 instance in your target DR Region and shut down the instance to save cost. You can then restart it and perform database restore and recovery as needed.

Alternatively, you can use automation such as AWS CloudFormation or Cloud Development Kit or third-party automation tools to launch an Amazon EC2 instance and install and configure SAP applications Oracle database when needed. Any automation must be created and tested in advance and we recommend performing frequent DR drills. This helps you save cost for Amazon EC2 instances and Amazon EBS volumes.

Note that the time to recover your database is dependent on the size of database. Any log files that were not copied to the DR Regions are lost and cannot be used for recovery. This option typically has higher RTO and RPO as compared to other options that use data replication technologies. However, it offers lower TCO in comparison to other options.



You can choose to deploy high availability and disaster recovery for the same production database instance.

If you want to use Oracle Data Guard for HA and DR, see Multiple Standby Databases Best Practices .

Sizing

Sizing applies to three key areas - compute, network, and storage.

Compute

AWS has certified multiple instance families with different sizes to run SAP workloads. For more details, see Amazon EC2 Instance Types for SAP.

To provision instances based on your requirements, you can use the Right sizing process. This process can help you optimize costs. Although it is ideal to use the right sizing approach when you move your SAP workloads to AWS Cloud, it is an ongoing process. We recommend you to use the latest generation of your selected instance family.

For a greenfield (new) deployment of SAP workloads, you can use the Quick Sizer tool to calculate the compute requirement in SAPS. This helps you to select the closest matching Amazon EC2 instance for a price that is most economical for you. Before completing your selection, ensure that the selected Amazon EC2 instance provides enough Amazon EBS and overall network throughput to meet your application requirements.

For migrations, you can use any of the following data sources to decide the right size of your instance:

- Source system utilization and workload patterns, such as EarlyWatch alert reports.
- Source system specification: CPU, memory, storage size + throughput + IOPS, network.
- Source system SAPS rating.

For more details, see Compute & storage.

Network

Network performance is often not explicitly stated as a requirement in SAP sizing. AWS enables you to check the network performance of all Amazon EC2 Instance Types.

Ensure that you have your network components setup to deploy resources related to your SAP workload. If you haven’t already setup network components like Amazon VPC, subnets, route tables etc., you can use the, AWS Quick Start Modular and Scalable VPC Architecture to most effectively deploy scalable Amazon VPC architecture in minutes. After setting up your Amazon VPC, you must set up Amazon EC2 instances within the Amazon VPC for your SAP workloads.

You also must set up a secured network connection between the corporate data center and the Amazon VPC along with the appropriate route table configuration, if it isn't already configured.

Storage

Deploying SAP workloads on AWS required a minimum storage size and layout, based on your choice of OS/DB platform. For further details, refer to the relevant SAP documentation. You need to create Amazon EBS volumes that match these requirements.

You must check that the storage required is enough to provide sufficient I/O performance. The new `gp3` volume is ideal for Oracle workloads that require smaller volume size. With `gp3`, the storage throughput and IOPS are decoupled from the size and can scale independently.

The io2 volume is well-suited for I/O-intensive database workloads that require sustained IOPS performance or more than 16,000 IOPS. The `io2 Block Express` is another provisioned IOPS SSD volume for workloads that require sub-millisecond latency, sustained IOPS performance, and more than 64,000 IOPS or 1,000 MiB/s of throughput.

For more details, see Storage for Oracle.

Amazon Machine Image (AMI)

You can deploy your SAP Oracle workload on Oracle Enterprise Linux 6.4 or later. A base AMI is required to launch an Amazon EC2 instance. You can create your own AMIs or obtain an Oracle Linux AMI from Oracle. For using AMIs from Oracle, see Launch an Oracle Linux instance in AWS. You can create your own Oracle Enterprise Linux image or use other images available at AWS Marketplace.

Security and compliance

The following are additional AWS security resources to help you achieve the optimum level of security for your SAP NetWeaver environment on AWS:

- AWS Cloud Security
- CIS AWS Foundations
- AWS Well-Architected Framework

OS Hardening

Check the following resources to strengthen the security of your workloads. You must have access to the SAP portal to view the SAP Notes.

- Refer to Security in Amazon EC2.
- Use Amazon Inspector.
- SAP Note 1635808
- SAP Note 2069760
- SAP Note 2936683
- SAP Note 1565179

To follow the CIS Benchmarks, see Securing Oracle Linux.

Encryption

The important aspect of securing your workloads is encrypting your data, both at rest and in transit. For more details, refer to the following:

- Amazon EBS encryption
- Data encryption in Amazon EFS
- Data encryption in Amazon S3

In addition to AWS encryption features, you can also use Oracle Transparent Data Encryption, as described in SAP Note 974876.

Security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. Security groups act at the instance level, not the subnet level.

Customers often separate the SAP system into multiple subnets, with the database in a separate subnet to the application servers, and other components, such as a web dispatcher in another subnet, possibly with external access.

If workloads are scaled horizontally, or high availability is necessary, you may choose to include multiple, functionally similar, Amazon EC2 instances in the same security group. In this case, you must add a rule to your security groups.

If Linux is used, some configuration changes may be necessary in the security groups, route tables, and network ACLs. For more information, see Security group rules for different use cases.

Network ACL

A network access control list (ACL) is an optional layer of security for your Amazon VPC that acts as a firewall for controlling traffic in and out of one or more subnets (they're stateless firewalls at the subnet level). You may set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your Amazon VPC.

See Amazon VPC Subnet Zoning Patterns for SAP on AWS to understand the network considerations for SAP workloads.

API call logging

AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you. The recorded information includes the identity of the caller, time of the call, source IP address, request parameters, and response elements returned by the AWS service. With CloudTrail, you can get a history of AWS API calls for your account, including API calls made via AWS Management Console, AWS SDKs, command line tools, and higher-level AWS services (such as, AWS CloudFormation). The AWS API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing.

Notification on access

You can use Amazon SNS or any third-party application to set up notifications on SSH login to your email address or mobile phone.

Storage for Oracle

This section describes the key considerations for designing storage layout of Oracle for SAP NetWeaver on AWS. Before defining the layout, we recommend familiarizing yourself with IOPS and throughput offered by Amazon EBS volume types and learning to calculate the baseline and burstable IOPS and throughput for these volumes. Amazon EC2 instances also have IOPS and throughput limits. For more details, see Amazon EBS-optimized instances.

Amazon FSx for NetApp ONTAP

FSx for ONTAP is certified for Oracle databases on SAP NetWeaver. For more information, see SAP Note 1656250 - SAP on AWS: Support prerequisites (portal access required).

File system

The file system structure for SAP Oracle deployment may differ with the database version. Refer to the following SAP Notes for individual Oracle database versions:

- SAP Note 2660017
- SAP Note 1915301
- SAP Note 1524205

The directory structure for database installation requires several file systems. This section only focuses on the storage layout of the file systems mentioned in the following table. The other file systems (used for storing Oracle software binaries, trace, and log files) are critical for operations but do not have heavy performance requirements as compared to the following files.

Description	File system
Database data files	/ oracle/<DBSID>/ sapdata(1,2….n)
Database online redo logs	/ oracle/<DBSID>/ origlog(A,B…)
Database mirror redo logs	/ oracle/<DBSID>/ mirrlog(A,B…)
Database offline redo logs	/ oracle/<DBSID>/ oraarch

You can calculate the capacity requirements from your existing database for migrations or using the SAP Quick Sizer tool for new implementations.

Calculate the IOPS

The most efficient way to estimate the actual IOPS that is necessary for your database is to query the system tables over a period of time and find the peak IOPS usage of your existing database. To perform this task, measure IOPS over a period of time and select the highest value.

You can access this information from the GV$SYSSTAT dynamic performance view, a special view in the Oracle database that provides performance information. The view is continuously updated while the database is open and in use. Oracle Enterprise Manager and Automatic Workload Repository reports access this view to gather data.

Alternatively, you can use the native storage tools to calculate the IOPS requirements. If storage tools are not available, you can use a script. For more information, see Determining the IOPS Needs for Oracle Database on AWS.

Amazon EBS volume types

AWS has multiple options for database storage, based on your throughput and IOPS requirements.

Two options for general purpose SSD:

- gp3 volumes deliver a consistent baseline rate of 3,000 IOPS and 125 MiB/s, included with the price of storage. You can provision additional IOPS (up to 16,000) and throughput (up to 1,000 MiB/s) for an additional cost.
- gp2 volumes deliver performance linked to the size of the volume. We recommend new customers to use gp3 volumes. Existing gp2 users can migrate to gp3 easily with Amazon EBS Elastic Volumes. It enables modification of volume types, IOPS or throughput of your existing Amazon EBS volumes without interrupting the Amazon EC2 instances.

Three options for provisioned IOPS SSD:

- `io1` is designed to deliver a consistent baseline performance of up to 50 IOPS/GB to a maximum of 64,000 IOPS, and provide up to 1,000 MiB/s of throughput per volume.
- `io2` is designed to deliver a consistent baseline performance of up to 500 IOPS/GB to a maximum of 64,000 IOPS, and provide up to 1,000 MiB/s of throughput per volume.
- `io2 Block Express` is designed for workloads that require sub-millisecond latency, sustained IOPS performance, more than 64,000 IOPS, and 1,000 MiB/s of throughput per volume.

Comparisons

Choosing between general purpose and provisioned IOPS SSD

We recommend using gp3 on Oracle for SAP on AWS workloads. It provides a better option for price over performance. You can dynamically switch from one volume type to another, if needed.

Choosing between `io1` *and* `io2`

We recommend you to use `io2` for provisioned IOPS use case. It provides lower annual failure rate and higher configurable IOPS per GB.

Choosing between `gp3` *and* `io2`

`io2` provides lower annual failure rate and higher maximum IOPS per volume but costs more than `gp3`. You can decide to use either of the two based your requirements regarding failure rate, IOPS, and cost.

Choosing between `io2` *and* `io2 Block Express`

`io2 Block Express` should be chosen over `io2` for workloads that require sub-millisecond latency and more than 64,000 IOPS and 1,000 MiB/s of throughput from a single volume. If `io2 Block Express` doesn't support your Amazon EC2 instance and Amazon EBS volume, use `io2`.

> **Note**
>
> Check the Amazon EBS volume types to ensure that your chosen volume supports the Amazon EC2 instance in use.

Best practices

Follow these best practices for maximizing your database performance and storage resilience:

- Use Amazon EBS-optimized instances for database storage. They have a dedicated path between Amazon EC2 and Amazon EBS volumes.
- To achieve higher IOPS and throughput, you can use Linux Volume Manager (LVM) to create Linux file systems with striping across multiple Amazon EBS volumes or Oracle Automatic Storage Management. For Automatic Storage Management, you can use multiple Amazon EBS volumes for creating disk groups, as recommended in the SAP installation guide for Oracle database.
- In case of backup to local file system, overall data throughput of the Amazon EBS volumes used to create the file system is crucial for backup performance.In AWS, you can use throughput optimized (`st1`) type Amazon EBS volumes for database backups to local file system. For larger file systems, `st1` type volumes have higher maximum throughput per volume at a lower cost when compared with `gp3` or `io1`/`io2`. Other volume types can be considered if `st1` doesn't meet your requirement. Ensure that the backup storage window can meet the required backup window available for database backups.

- When running SAP NetWeaver on Oracle database, you are required to have the `/sapmnt/[SID]` directory mounted on the database server. We recommend that you use Amazon EFS to host the `/sapmnt` directory and mount this on all SAP servers using the NFS protocol.
- Data and log files should be on separate volumes.
- Origlog and mirrlog should be on separate volumes.
- Copies of control files should be stored on file systems that are created on separate volumes.
- Redo log files are written sequentially by the Oracle database instance Log Writer (LGWR) process. Log file systems must be designed to support such I/O activity.

Example configuration

You need to set up the Oracle database with the following requirements:

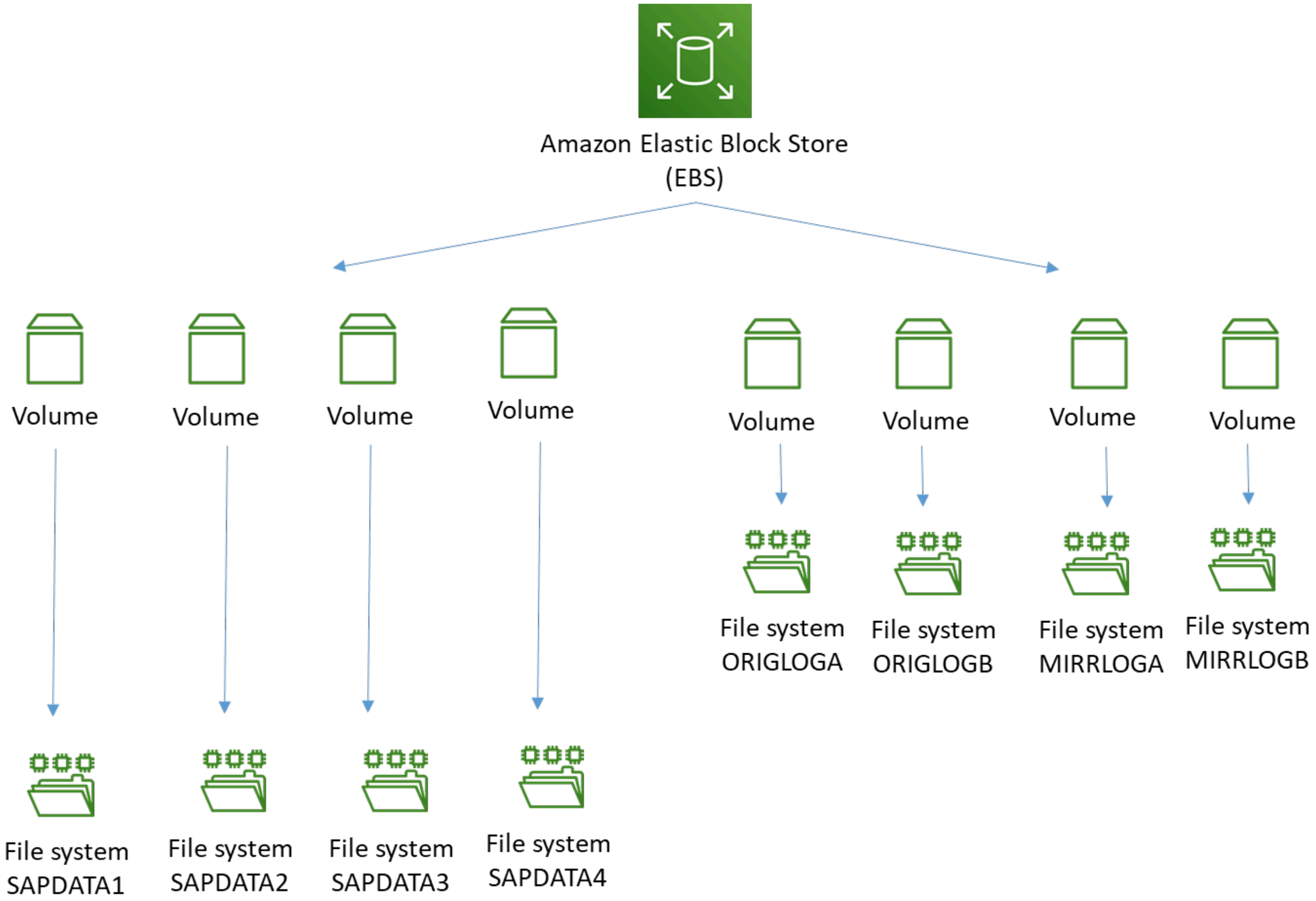
- Data file system with 48,000 peak IOPS, 3,000 MiB/s throughput and 12 TB capacity
- Each of the archive log, online and mirror redo log files with 3,800 peak IOPS and 25 GB capacity
- Oracle based file system for all other directories under `/oracle` with 300 GB capacity

The following is an example storage design to achieve the previously mentioned performance and capacity requirements.

Data file system with provisioned IOPS using gp3 or `io2` and without LVM:

Create one Amazon EBS volume for each file system size and use provisioned IOPS and throughput, as per the individual file system requirements. For data volumes, higher size and IOPS can be assigned to application tablespace volumes as needed. Since size, type, and IOPS can be changed dynamically, you can adjust these parameters with changing requirements.

This is a simpler approach and doesn’t require any volume striping using Linux LVM or similar technology. However, with this approach, you are still limited by the maximum size and IOPS supported by individual Amazon EBS volumes.



Data file system with provisioned IOPS using gp3 or `io2` and with LVM:

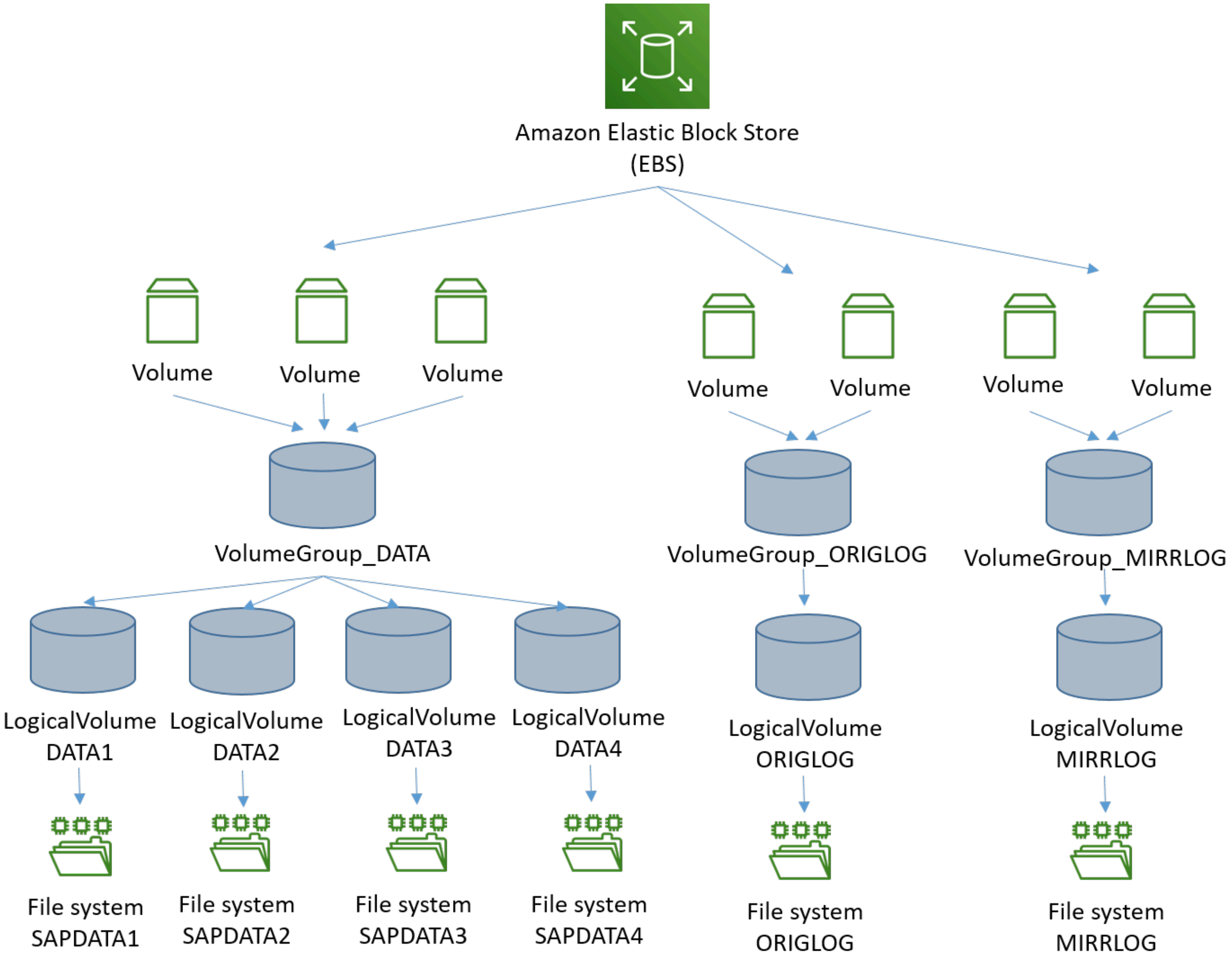
Create a single data volume group using stripping and four Amazon EBS volumes of 3 TB/each. This provides 12 TB of file system capacity. Each data volume can be configured with 12,000 IOPS and 750 MiB/s throughput to provide combined IOPS of 48,000 and throughput of 3,000 MiB/s. Multiple `sapdata` logical volumes and file systems can be created under this volume group. You can also create multiple volume groups for data file systems alone.

Creating volume groups increases the recovery time, in case one of the underlying Amazon EBS volumes becomes unusable. In this case, many data files residing on this volume group may be impacted as all `sapdata` file systems are stripped across all Amazon EBS volumes.

The benefit of using this approach is that you don't need to have the IOPS and throughput requirements of individual `sapdata` file systems. The requirements of the Oracle database data file system are sufficient. Also, `sapdata` shares the IOPS and throughput, leading to better utilization of baseline performance when using gp3 volumes.

Online redo logs

Use 25 GB gp3 Amazon EBS volumes with 3,800 provisioned IOPS.



Deployment

- Standalone deployment
- HA/DR deployment

Standalone deployment

In this example, we set up a sample environment for installation. It includes a public subnet for RDP and SSH access via the internet. We use AWS Launch Wizard for SAP in a single Availability

Zone deployment to create the Amazon VPC, subnets, security groups, and IAM roles. You can refer to this example set up but should also follow your own network layout and comply with security standards, such as the following:

- Using a Landing Zone solution like AWS Control Tower.
- Working with a cloud team like Cloud Center of Excellence to use existing standards.

Step 1: Prepare your AWS account

Check the Region where you want to deploy your AWS resources:

- You pick your region for deployment during the planning phase.
- Display the AWS Command Line Interface configuration data:

```
$ aws configure list
```

Ensure that the default region listed in the command output is the same as the target region where you want to deploy your AWS resources and install SAP workloads. In this deployment, we provision an Amazon EC2 instance.

> **Note**
>
> In this section, the syntax used for the AWS CLI and Linux commands is specific to the scope of this document. Each command supports many additional options. To learn more, use the `aws help` command.

Step 2: Create a JSON file for Amazon EBS storage

Create a JSON file containing the storage requirements for SAP Oracle database server volumes. The following is an example `JSON` file with two Amazon EBS volumes for swap and installation directories. You can add more volumes as per your storage design.

```
[
  {
    "DeviceName": "/dev/sdh",
    "Ebs": {
      "VolumeSize": 32,
      "VolumeType": "gp3",
      "DeleteOnTermination": true
    }
  },
  {
    "DeviceName": "/dev/sdg",
    "Ebs": {
      "VolumeSize": 50,
      "VolumeType": "gp3",
      "DeleteOnTermination": true
    }
  }
]
```

In the preceding example, the device name `/dev/shd` is for non-nitro based hypervisors. On Nitro-based instances, Amazon EBS volumes are presented as NVME block devices. You need to perform additional mapping when configuring these volumes. For more information, see Operating system and storage configuration.

Step 3: Launch the Amazon EC2 instance

Launch the Amazon EC2 instance for the SAP Oracle database installation in your target region, using the information gathered in Step 1. You must create the required storage volumes and attach them to the Amazon EC2 instance for the SAP installation, based on the `JSON` file you created in the Amazon EBS storage (Step 2).

```
$ aws ec2 run-instances \
--image-id <AMI-ID> \
--count <number-of-EC2-instances> \
--instance-type <instance-type> \
--key-name=<name-of-key-pair>
--security-group-ids <security-group-ID> \
--subnet-id <subnet-ID> \
--block-device-mappings file://C:\Users\<file>.json \
--region <region-ID>
```

Use this command in a single line format, as shown in the following example.

```
aws ec2 run-instances --image-id ami-xxxxxxxxxxxxxxx --count 1 --instance-type m5.large
 --key-name=my_key --security-group-ids sg-xxxxxxxx --subnet-id subnet-xxxxxx  --block-
device-mappings file://C:\Users\<file>.json
```

You can also launch Amazon EC2 instances using the AWS Management Console. For more information, see Launch an instance.

Step 4: Prepare the Oracle Linux OS

Before starting the installation, you need to perform Oracle Enterprise Linux specific prerequisite tasks. For more information, refer to SAP Notes 1635808, 2069760, and 2936683 (login required).

Step 5: Prepare each Amazon EC2 instance for SAP Oracle installation

Download the SAP installation media as per the SAP installation guide, for the version of SAP NetWeaver you want to install on your Amazon EBS volumes. Locate your installation guide on the Guide Finder for SAP NetWeaver.

If you choose to install the SAP Oracle database with high availability deployment across two Availability Zones, repeat the preceding steps for Oracle database standby HA instance in the second Availability Zone.

If you choose to install SAP Oracle database with high availability and disaster recovery deployment across two AWS Regions, repeat the preceding steps in the second AWS Region in which you want to run the Oracle database standby DR instance.

Step 6: Installing SAP Oracle on Amazon EC2 instances

You are now ready to install the SAP Oracle software on your Amazon EC2 instances. For more information, see the Oracle Database Software Installation section of your SAP NetWeaver installation guide. Locate your installation guide on the Guide Finder for SAP NetWeaver. Instructions are available for all supported Oracle database versions.

HA/DR deployment

Installing SAP Oracle on Amazon EC2 instances and configuring HA/DR

Create an additional Amazon EC2 instance and perform the installation in a secondary Availability Zone. The steps for creating a HA or DR instance in a secondary Availability Zone are the same as described in Standalone deployment. You can simplify this step by using the following methods.

- If you have built any automation using AWS CloudFormation or other tools to create the primary Amazon EC2 instance and install database software, you can use the same automation to build the HA instance.

- You can create an Amazon Machine Image of the primary Amazon EC2 instance and launch another instance in the secondary Availability Zone.

The configuration of high availability or disaster recovery depends on the tools you use. See the next sections for more details.

> **Note**
>
> The preceding steps are not applicable to passive DR.

Third-party references

To configure the SAP Oracle system with HA/DR using the Oracle Data Guard, refer to the following documents.

- Setting up Oracle 12c Data Guard for SAP
- Setting up Oracle 12c Data Guard for SAP
- Oracle Standby Databases
- Configuring Oracle Data Guard

For information about configuring HA/DR using a third-party product, see the vendor-specific documentation, such as the following.

- SIOS Oracle High Availability
- Veritas InfoScale™ 7.4.1 Solutions

> **Note**
>
> You need to configure cross-regional Amazon VPC peering or Transit Gateway to enable SAP Oracle asynchronous replication between the two Regions.

To perform a manual failover or switchover, see HA/DR operations.

Operations

Tagging AWS resources

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value, both defined by you. Adding tags to various AWS resources will make managing SAP environments more efficient, and help you search for resources quickly. Many Amazon EC2 API calls can be used in conjunction with a special tag filter. For more information, see Tagging AWS resources. The following are some examples of how you can use tags for your operational needs.

- You can tag your Amazon EBS volumes to identify their environment and use the same tags to create backup policies. For instance, Environment=DEV/QAS/PRD.
- You can use similar tags (DEV/QAS/PRD) for Amazon EC2 instances and use them for patching your OS or running scripts to stop/start applications or Amazon EC2 instances.

Monitoring

AWS provides multiple native services to monitor and manage your SAP environment. CloudWatch and CloudTrail can be used to monitor your underlying infrastructure and APIs. CloudWatch provides ready-to-use KPIs for CPU, disk utilization, and enables you to create custom metrics for KPIs that you want to monitor. CloudTrail allows you to log the API calls made to your AWS infrastructure components.

Operating system maintenance

In general, operating system maintenance across large estates of Amazon EC2 instances can be managed by using:

- Tools specific to the operating system, like Oracle Enterprise Manager
- Third-party products, such as those available on AWS Marketplace.
- AWS Systems Manager

The following are some key operating system maintenance tasks

Patching

You can follow SAP recommended patching process to update your landscape on AWS. With AWS Systems Manager Patch Manager, you can roll out OS patches according to your corporate policies. It has multiple benefits:

- Scheduling based on tags
- Defining patch baselines
- Auto-approving patches with lists of approved and rejected patches

AWS Systems Patch Manager integrates with IAM, CloudTrail, and CloudWatch Events to provide a secure patching experience that includes event notifications and the ability to audit usage. For details about the process, see How Patch Manager operations work. Third-party products are available on AWS Marketplace.

Maintenance Windows

AWS Systems Manager Maintenance Windows lets you define a schedule to perform potentially disruptive actions on your instances, such as patching an operating system, updating drivers, installing software or patches.

Administrator access

For administrative purposes, you can access the backend of your SAP systems via SSH or AWS Systems Manager Session Manager.

Automation

AWS Systems Manager Automation simplifies common maintenance and deployment tasks of Amazon EC2 instances and other AWS resources. For more information, see AWS Systems Manager Automation.

Automation using Infrastructure-as-Code with AWS CloudFormation

We recommend following the principle of Infrastructure-as-Code (IaC) for automating and maintaining your workloads on AWS. AWS CloudFormation provides a common language for you to describe and provision all the infrastructure resources in your cloud environment in a repeatable and automated manner.

Cost optimization

We recommend cost optimization as an ongoing process. There are many AWS services that help with budgeting, cost control and optimization. For more details, see Cost Optimization Pillar - AWS Well-Architected Framework

Compute & storage

Compute

Amazon EBS volumes are exposed as NVMe block devices on Nitro-based instances. When changing Amazon EC2 instance types from a previous generation to a Nitro generation, NVMe device IDs associated with the volume can change. To avoid mount errors during change of instance type or instance reboots, you need to create a label for your file systems and mount it by the label, *and not* the NVMe IDs. For more details, see support article.

Aside from operating system maintenance, you should consider maintenance for your Amazon EC2 instances. It can be driven by using Automation runbook. The following are some examples.

- Use `AWS-StopEC2InstanceWithApproval` to request one or more IAM users approve the instance stop action. After the approval is received, runbook stops the instance.
- Use `AWS-StopEC2Instance` to automatically stop instances on a schedule, using CloudWatch Events or a Maintenance Window task. For example, you can configure an Automation workflow to stop instances every Friday evening and restart on Monday mornings. Note that this automation will only stop and start the Amazon EC2 instance. You must create additional document to gracefully stop and start SAP applications and database and then use the AWS Systems Manager to run such automations.
- Use `AWS-UpdateCloudFormationStackWithApproval` to update resources that were deployed using AWS CloudFormation template. The update applies a new template. You can configure the Automation to request approval by one or more IAM users before the update begins.

You can also use AWS Instance Scheduler to configure custom start and stop schedules for Amazon EC2 and Amazon RDS instances.

Storage

The following are the storage services used across this guide.

- Amazon EBS provides persistent storage for SAP applications and database. Amazon EBS volumes can be resized and even have the volume type changed without disrupting the applications. For more details, see Amazon EBS Elastic Volumes. After modifying the Amazon EBS volume, you need to extend the file system to match the extended volume size. For more details, see Extend a Windows file system after resizing a volume.
- Amazon EFS does not require you to explicitly provision storage, you pay only for your usage. It is built to scale on demand, without disrupting applications, growing and shrinking automatically as you add and remove files. This ensures that your applications have the required storage.
- Amazon S3 also does not require you to explicitly provision storage, you pay only for your usage. You can use Object lifecycle management to set rules that define when objects are transitioned or archived to colder storage (Amazon S3 IA or S3 Glacier) and when they expire. For more information, see Managing your storage lifecycle.

Backup & restore

Snapshots and AMIs

A common approach for backing up your SAP NetWeaver application servers is using snapshots and AMIs.

The SAP application data is stored on Amazon EBS volumes attached to the SAP NetWeaver application servers. You can backup the data on these volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups of Amazon EBS volumes, which means that only the blocks on the device that have changed after your most recent snapshot are saved. For more information, see Create Amazon EBS snapshots.

An Amazon Machine Image (AMI) provides the information required to launch an instance along with a block device mapping of all Amazon EBS volumes attached to it.

Amazon EC2 powers down the instance before creating the AMI to ensure that everything on the instance is stopped and in a consistent state during the creation process. If you’re confident that your instance is in a consistent state appropriate for AMI creation, you can check the *No Reboot* option.

You can use AWS Backup to centrally configure backup policies and monitor backup activity for these snapshots. Once you have completed the SAP installation and post installation steps, create an image of the instance.

+

```
aws ec2 create-image --instance-id i-1234567890abcdef0 --name "My server" --description
 "An AMI for my server"
```

AWS provides a very simple and quick way to copy an SAP system. You can use the AWS Console Home or the AWS CLI to create a new AMI of an existing SAP system. You can then launch exact copies of the original system from the new AMI. For more details, see Amazon Machine Images (AMI).

Backup to Amazon S3

You can perform traditional file-based backup to Amazon S3 from your Amazon EBS volumes. One way to take backup is to use AWS CLI and initiate it by using AWS Systems Manager Run command, so that you can centrally manage the backups.

Backup with third-party products

There are many third-party products for AWS services, including a number that have been certified by SAP. Go to SAP Partner Services and Solutions Directory, select **ISV Solutions** in *Service/Solution Type*, then **Backup and Recovery** in *Software Solution*.

Amazon EFS backup

Using AWS Backup, you can centrally configure backup policies and monitor backup activity for AWS resources, including Amazon EFS file systems.

Alternatively, you can perform a file-level backup of your Amazon EFS file system to Amazon S3. You can do this by running a file-level copy to Amazon S3 from any Amazon EC2 instance running in the same region. This can be automated and scheduled using AWS Systems Manager Run Command in combination with CloudWatch Events.

Backup and restore for Oracle database

You must to regularly backup your operating system and database to recover them in case of any failure. AWS Cloud provides various services and tools that you can use to backup your Oracle database.

Storage snapshots

You can backup your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only blocks on the device that have changed after your most recent snapshot are saved. Snapshots of Amazon EBS volumes can be created for backup of Oracle database file systems like Oracle home and stage directories.

For complete Oracle database file backups using snapshots, you can use the Storage Snapshots Optimization feature by Oracle, supported from version 12c. For more details, see Making Backups with Third-Party Snapshot Technologies. AWS customers can use this feature in combination with Amazon EBS snapshots to perform Oracle database backups. Using snapshot backups may reduce the backup window as compared to the full database backup approach. To set up snapshot-based based database backup, you can use sample scripts and steps published in the blog Improving Oracle backup and recovery performance with Amazon EBS multi-volume crash-consistent snapshots.

Oracle database backups

Any one of the following methods can be used for Oracle database backup.

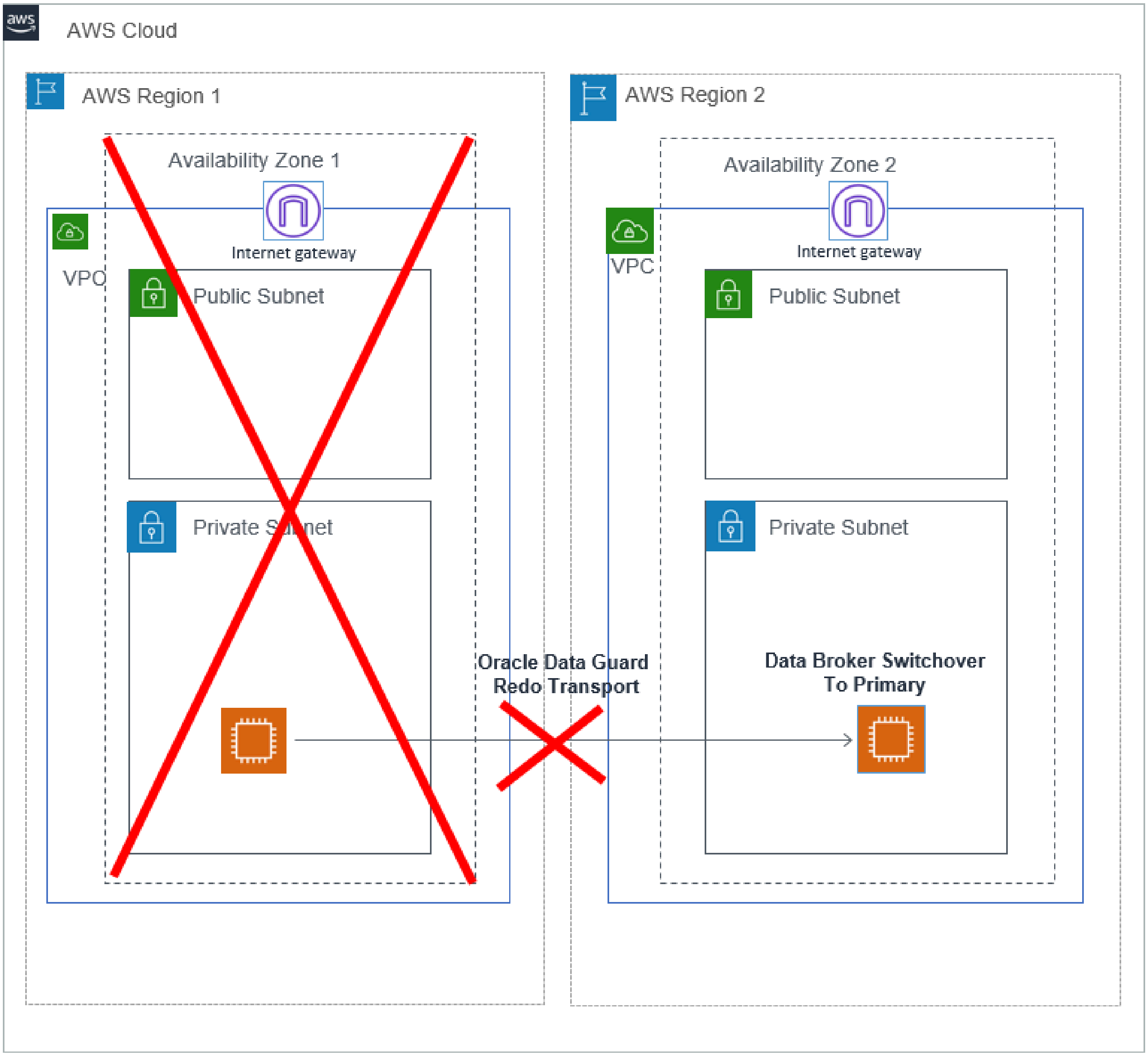
- Oracle database native tools (BRTOOLS) can be used to take backups on local storage. Once the backup is complete on local storage, it can be moved to Amazon S3 bucket via scripts.
- Oracle Secure Backup Cloud Module to backup your database directly to Amazon S3 using RMAN. For setup, see Oracle Database Backup To Cloud: Amazon Simple Storage Service (S3). For licence requirements, see Oracle Secure Backup Licensing Information.
- You can backup your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. For more information, see the preceding Storage snapshots section.
- There are many third-party tools from partner like Commvault, NetBackup, etc. that use the SAP backint interface and have Oracle database agents, with the capability to backup the database directly to Amazon S3.

To configure and tune backups for your Oracle database, see SAP on Oracle – Backup and Recovery and Database Backup and Recovery User’s Guide - Tuning RMAN Performance.

HA/DR operations

Oracle Data Guard

You can use manual failover or switchover to the standby database using steps described in the Switchover and Failover Operations. You can also automate this process by following the steps in Oracle Data Guard Fast-Start Failover. When using this feature with the observer node, you must place the observer node in the third Availability Zone. Ensure that you have the license to use the fast-start failover, it may not be included with the Data Guard. You can also use supported third-party products that provide automatic failover operation with the Data Guard. To reconnect the SAP applications post-failover, see the *Reconnect SAP instance to database* section-premises in https://www.sap.com/documents/2016/12/a67bac51-9a7c-0010-82c7-eda71af511fa.html.



Perform a DNS change

In case of manual failover, you may install SAP application servers using a virtual hostname and perform a DNS change to direct the SAP application servers to the new primary database server. For a DNS resolution in AWS, you can use any of the following options.

- Amazon Route 53 enables you to create a private hosted zone for your environment and an A record for the virtual hostname used for Oracle database. Initially, this A record is mapped to the IP address of the primary Oracle database instance.
- You can maintain your own DNS server on-premise or on your Amazon EC2 instances. You can create an A record there for your virtual hostname used for Oracle database. Initially, this A record is mapped to the IP address of the primary Oracle database instance.
- With the AWS Directory Service, you can create an A record for the virtual hostname used for Oracle database.

With any of the previously mentioned options, you can change the A record to a private IP address of the primary database instance in case of a failover. This DNS change can also be automated using AWS services and scripts.

Resources

SAP on AWS customers have the flexibility to deploy SAP Oracle database on the scalable, on-demand Amazon EC2 platform in a highly available manner. They don’t have to invest in costly capital expenditures for the underlying infrastructure. By combining the AWS platform flexibility and SAP installation techniques, our customers greatly improve the availability of their deployments. For more details, see SAP on AWS Case Studies.

Support

AWS offers two levels of support. AWS Business Support provides resources and technical support for customers running SAP workloads on AWS. AWS Enterprise Support offers support to customers running mission critical SAP production workloads on AWS.

Document revisions

Date	Change
December 2021	Initial publication

SAP ASE for SAP NetWeaver on AWS Deployment and Operations Guide for Linux

This guide provides information about configuring SAP ASE database for SAP NetWeaver on AWS.

Prerequisites

The following information is required to deploy SAP Adaptive Server Enterprise (ASE) for SAP NetWeaver applications on AWS. This pertains to your existing resources, using AWS CLI to create Amazon EC2 and Amazon EBS resources.

Information

Information	Description
AWS Region	Region where you want to deploy your AWS resources.
Availability Zone (AZ)	Availability Zone within your target Region where you want to deploy your resources.
Amazon VPC id	Amazon VPC where you want to deploy your Amazon EC2 instances for SAP installation.
VPS subnet id	Subnet where you want to deploy your Amazon EC2 instances.
Linux AMI id	Amazon Machine Image (AMI) that will be used to launch your Amazon EC2 instances . You can find the latest Linux AMIs on AWS Marketplace.
Key pair	Make sure that you have generated the key pair in your target Region and that you have access to the private key.
Security group id	Name of the security group that you want to assign to your Amazon EC2 instances.

Access key ID	Access key for your AWS account that will be used with AWS CLI tools.
Secret access key	Secret key for your AWS account that will be used with AWS CLI tools.

- Create security groups and open ports to enable communication. For existing security groups, ensure that the required ports are open. For a list of ports, refer to TCP/IP ports of all SAP products.
- Ensure that you have installed and configured AWS CLI with required credentials, if you plan to use it to launch instances. For more information, see Installing the AWS CLI.
- If you plan to use the AWS Management Console, ensure that you have the essential credentials and permissions to launch and configure AWS services. For more information, see Access management for AWS resources.
- Ensure that you have the software files required for installation readily available. You can stage these in Amazon S3 or Amazon Elastic File System (Amazon EFS). Amazon EFS can be easily shared on all of your installation hosts. For more information, see Create your Amazon EFS file system.
- You can request a service limit increase by creating a support ticket. For more information, see AWS service quotas.

References

You can refer to the following resources before deploying SAP ASE on AWS. If you are new to AWS, see Get started with AWS.

- What is Amazon EC2?
- Amazon Elastic Block Store (Amazon EBS)
- What is Amazon S3?
- What is Amazon VPC?
- What is IAM?
- SAP on AWS Overview and Planning
- SAP Lens - AWS Well-Architected Framework

- Storage options for Oracle Database

  The storage options for Oracle are also valid for ASE.
- Performance and Tuning Series: Basics
- Installation of SAP Systems Based on the Application Server ABAP of SAP NetWeaver 7.3 EHP1 to 7.52 on UNIX: SAP Adaptive Server Enterprise
- Installation of SAP Systems Based on the Application Server Java of SAP NetWeaver 7.5 and SAP Solution Manager 7.2 SR2 Java of SAP NetWeaver 7.5 on UNIX: SAP Adaptive Server Enterprise
- SAP Note 2922454 - SAP Adaptive Server Enterprise (SAP ASE) on Cloud Platforms (requires SAP portal access)
- SAP Note 1941500 - Certification information for Linux and other Operating Systems - SAP ASE (requires SAP portal access)

Planning

Plan your SAP system landscape according to the SAP Master Guide for your version of SAP system running ASE on Linux. We recommend referring to the following SAP Notes (require SAP portal access).

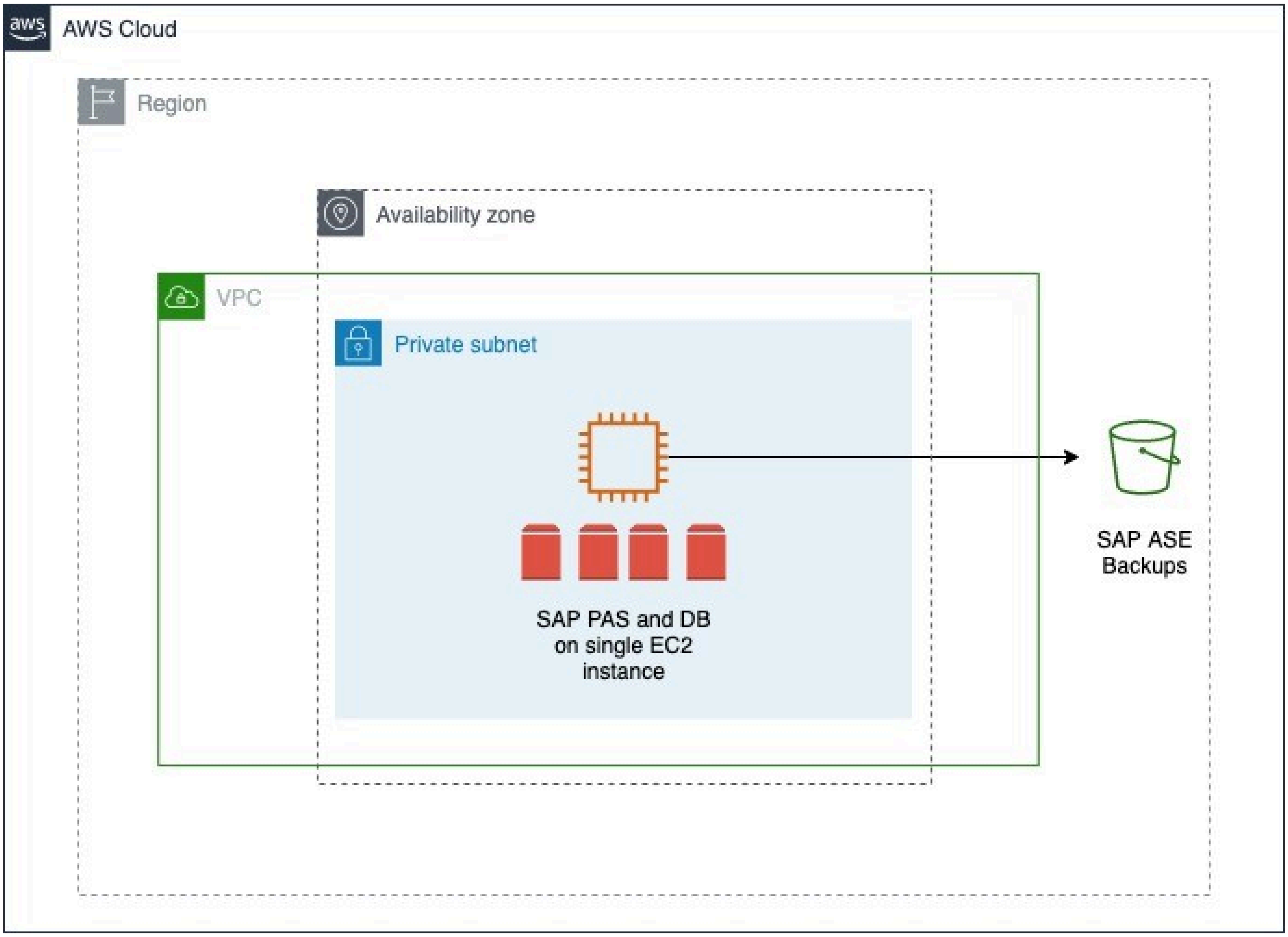
- SAP Note 1748888 - SYB: Inst.Systems Based on NW 7.3 and Higher: SAP ASE
- SAP Note 1554717 - SYB: Planning information for SAP on SAP ASE
- SAP Note 1656250 - SAP on AWS: Support prerequisites

Deployment options

To install SAP ASE for SAP NetWeaver, you have four deployment options:

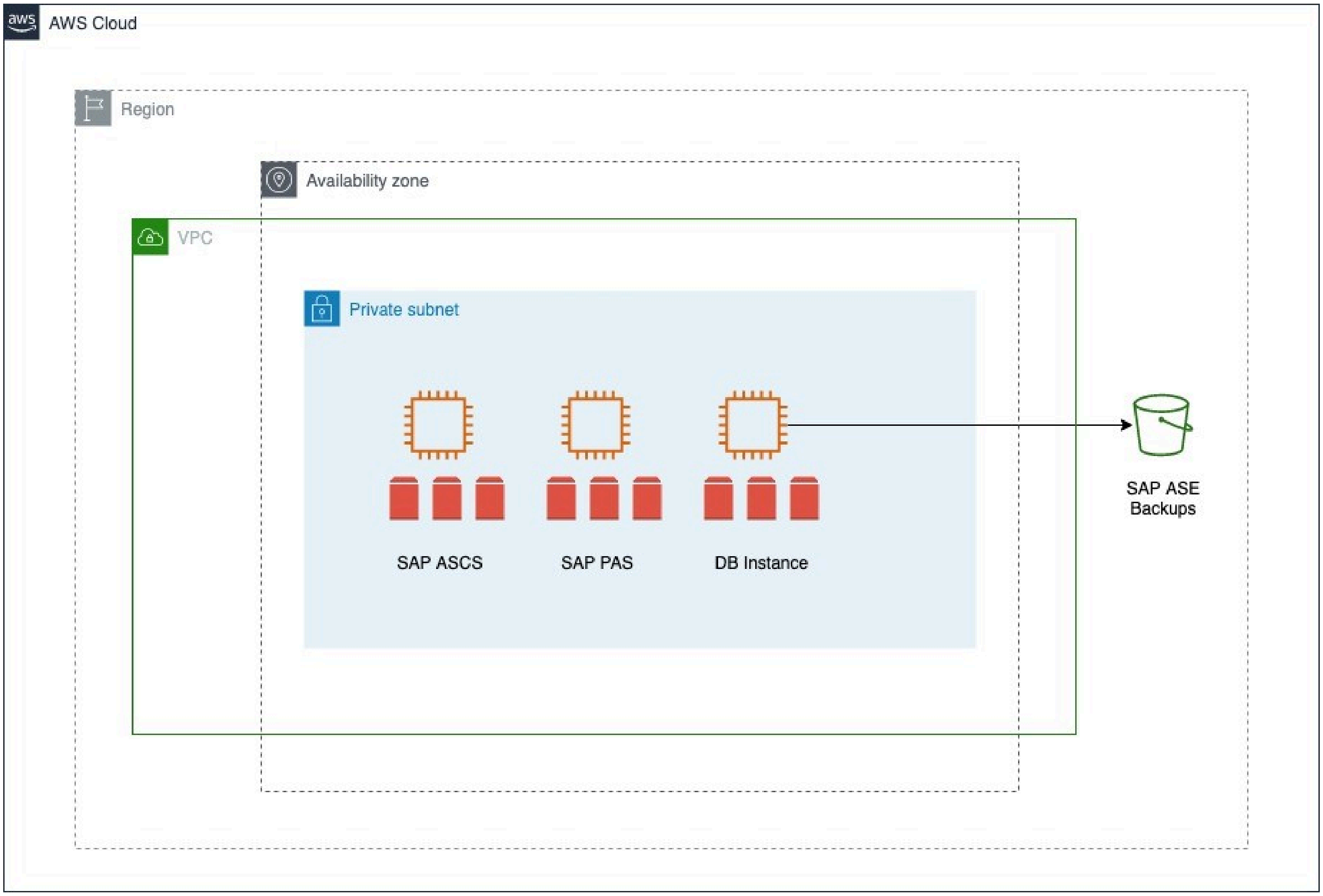
Standalone deployment

In standalone deployment (also known as single host installation), all components of the SAP NetWevaer, ABAP SAP Central Services (ASCS), and database Primary Application Server (PAS) run on one Amazon EC2 instance using a single Availability Zone in an AWS Region. This option is recommended for non-production workloads. You can use Amazon EC2 auto recovery feature to protect your instance against infrastructure issues like loss of network connectivity or system power. This solution is not database state aware, and does not protect your database against storage failure, OS issues, Availability Zone or Region failure.



Distributed deployment

In distributed deployment, every instance of SAP NetWeaver (ASCS/SCS, database, PAS, and optionally AAS) can run on a separate Amazon EC2 instance. This system also deploys SAP ASE database in a single Availability Zone. You can use Amazon EC2 auto recovery feature to protect your instance against infrastructure issues like loss of network connectivity or system power. This solution is not database state aware, and does not protect your database against storage failure, OS issues, Availability Zone or Region failure.

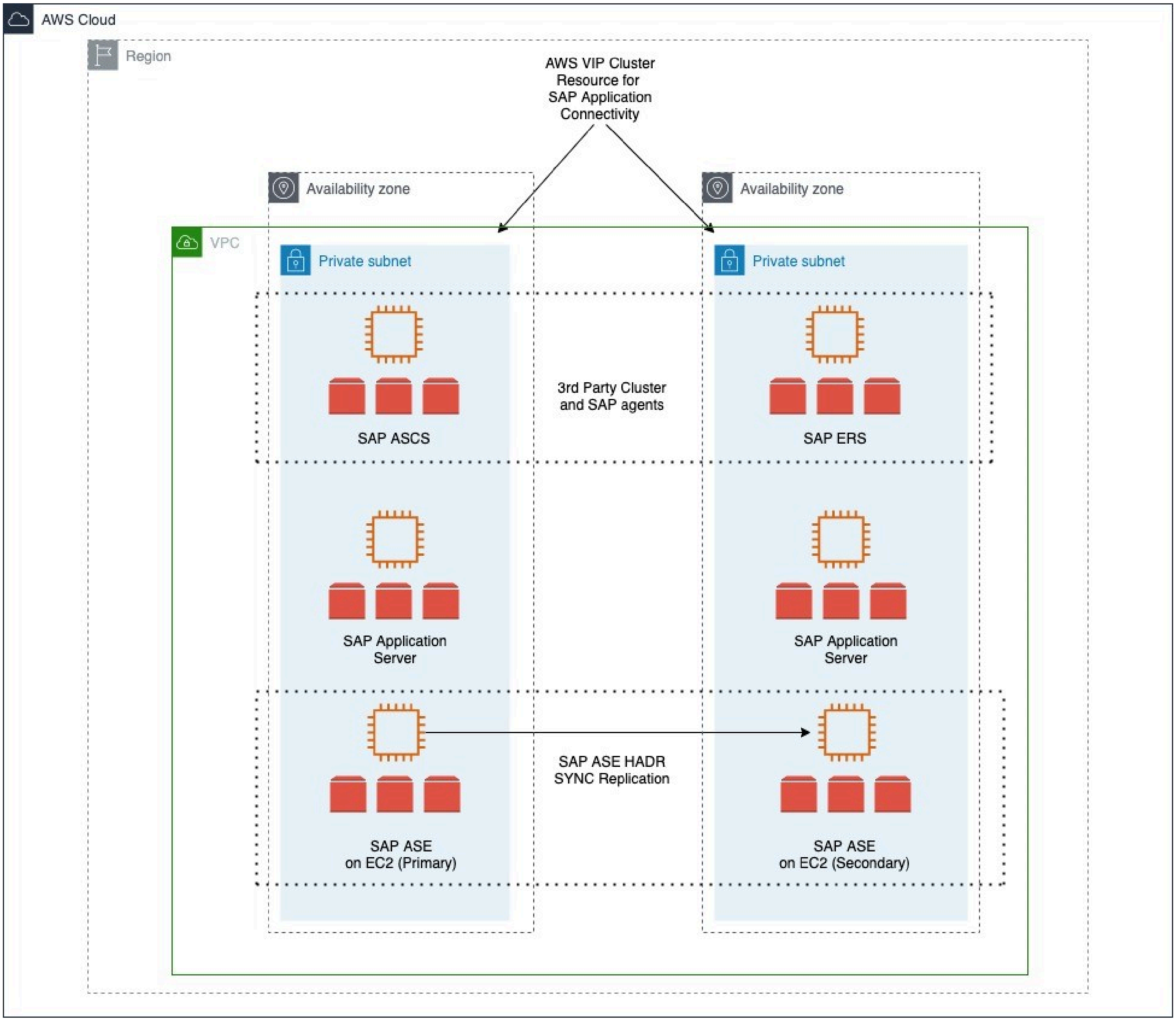


High availability deployment

In a high availability deployment, you deploy two Amazon EC2 instances across two Availability Zones within a Region, and the SAP ASE database with a combination of the Data Movement Component and Database Fault Manager.

All Availability Zones within an AWS Region are connected with high-bandwidth over fully redundant and dedicated metro fiber, providing high-throughput and low-latency networking between Availability Zones. For a high availability configuration, you can set up a primary and standby relationship between two SAP ASE databases with synchronous replication, running on Amazon EC2 instances in different Availability Zones within the same Region.

The SAP ASE always-on option is a high availability/disaster recovery system that contains two or more SAP ASE servers – the primary server where all of the transaction processing takes place, and the warm standby (companion) server. The primary and standby nodes are deployed in different Availability Zones, providing protection against zonal failures. You can also integrate Fault Manager to automatically failover the system in case of failures. You can learn more about this on the SAP Help Portal’s SAP Adaptive Servers Enterprise HADR Users Guide.



We recommend referring to the following SAP Notes (SAP portal access required) for a high availability deployment.

- *SAP Note 1650511 – SYB: High Availability Offerings with SAP Adaptive Server Enterprise*
- *SAP Note 2808173 – Special Instructions when Installing and Upgrading HADR with SAP Business Suite on SAP ASE*

Disaster recovery deployment

You can increase business continuity with disaster recovery deployment of your SAP systems on AWS Cloud. Based on recovery time objective, recovery point objective, and cost, you can set up disaster recovery deployment with one of the following three options.

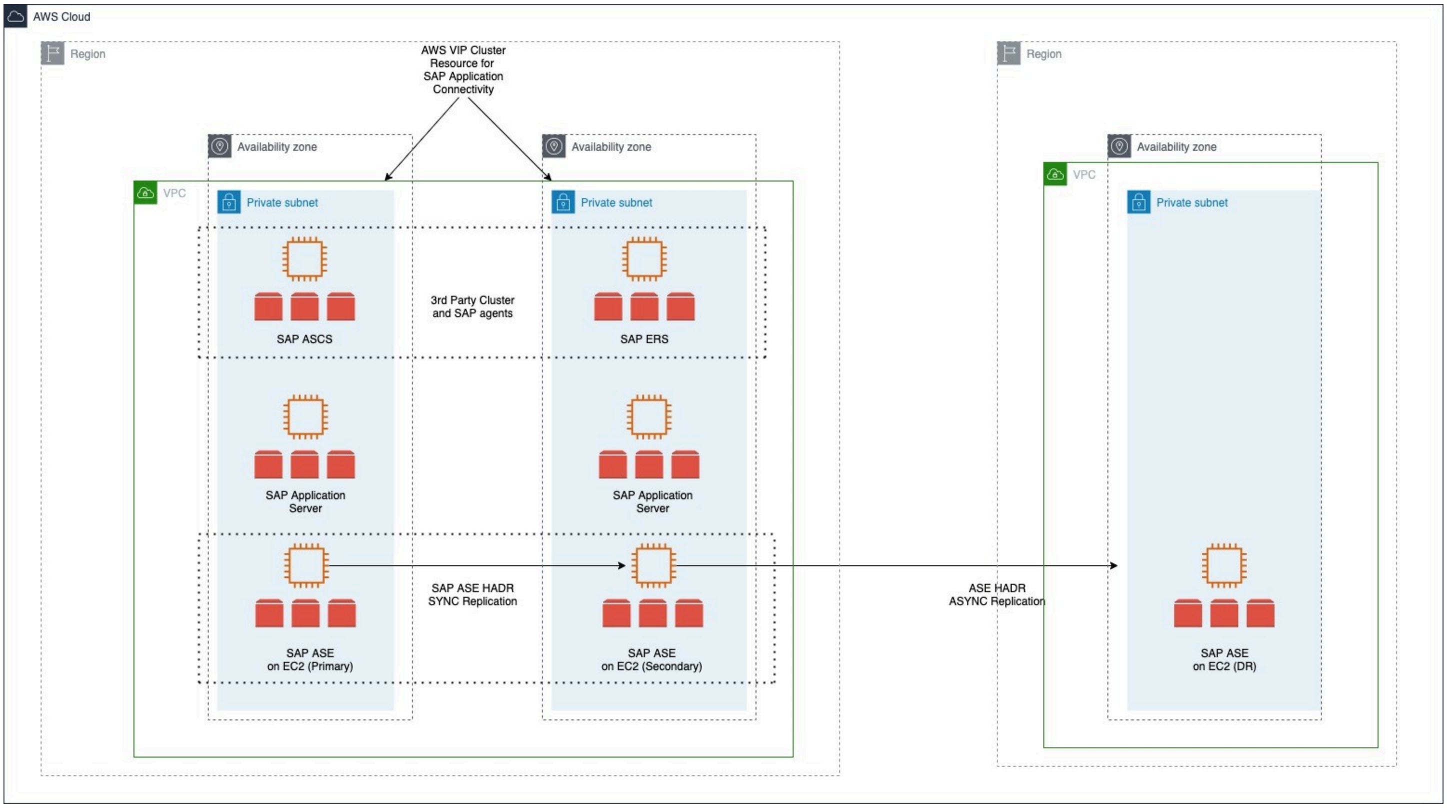
Option 1 – disaster recovery using the SAP ASE HADR feature

You can use the SAP ASE HADR feature to replicate your database in a secondary AWS Region or Availability Zone, based on your business and audit requirements. You can also integrate this DR node in an existing HADR landscape. This setup enables you to increase the overall system resiliency.

With this option, you can either choose pilot light, where the recovery instance is smaller than the current instance, or hot standby, where the recovery instance is of the same size as the current instance. You must consider your recovery time objectives and manual effort required when choosing between pilot light or hot standby. The recovery instance for the pilot light option must be resized before assuming disaster recovery.

For more details, check the following SAP resources.

- *HADR System with DR Node Users Guide*
- *2934459 - HADR support of two ASE servers on Primary and Companion machines - SAP ASE (requires SAP portal access)*

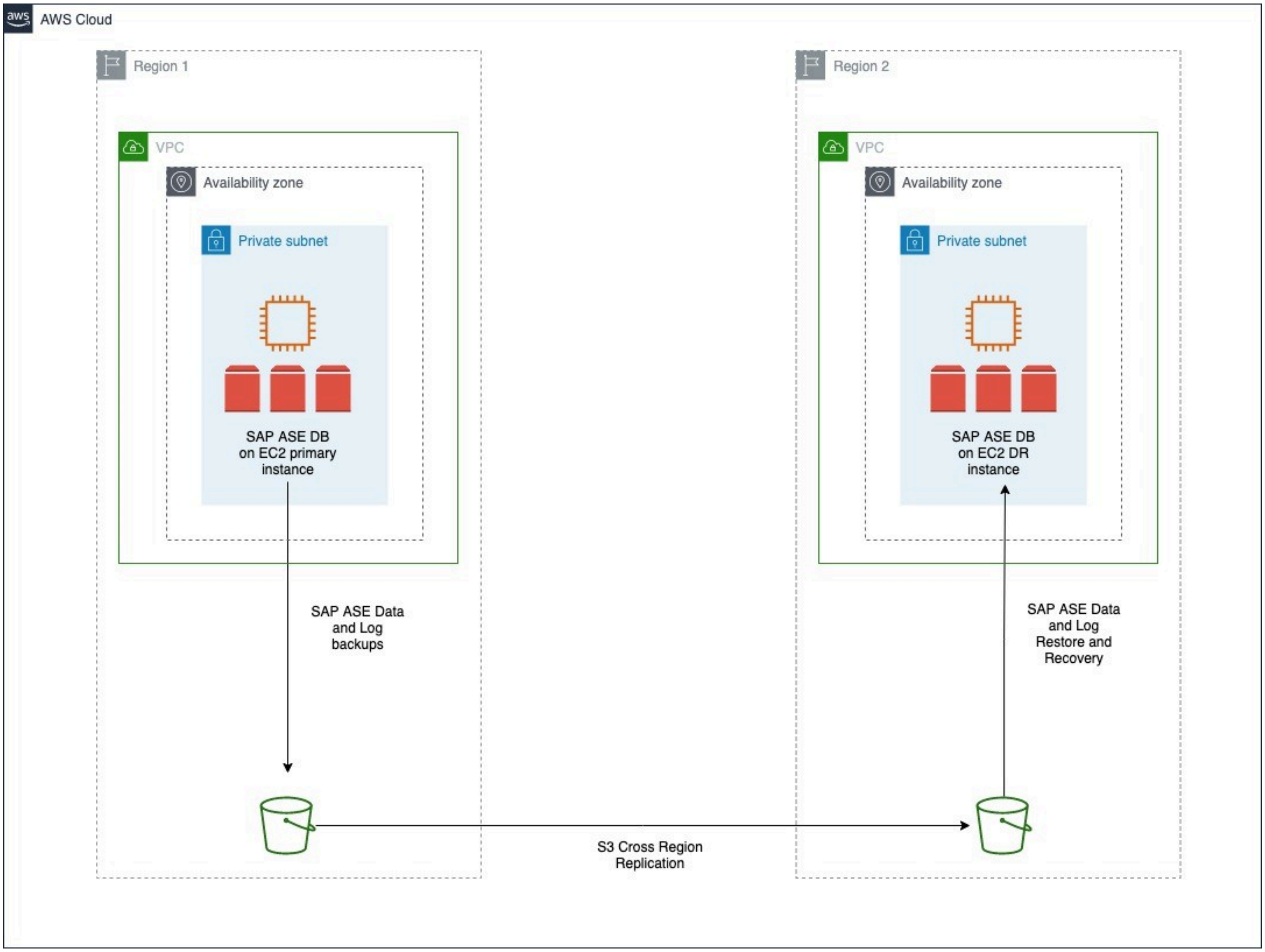


Option 2 – passive disaster recovery using backup and recovery

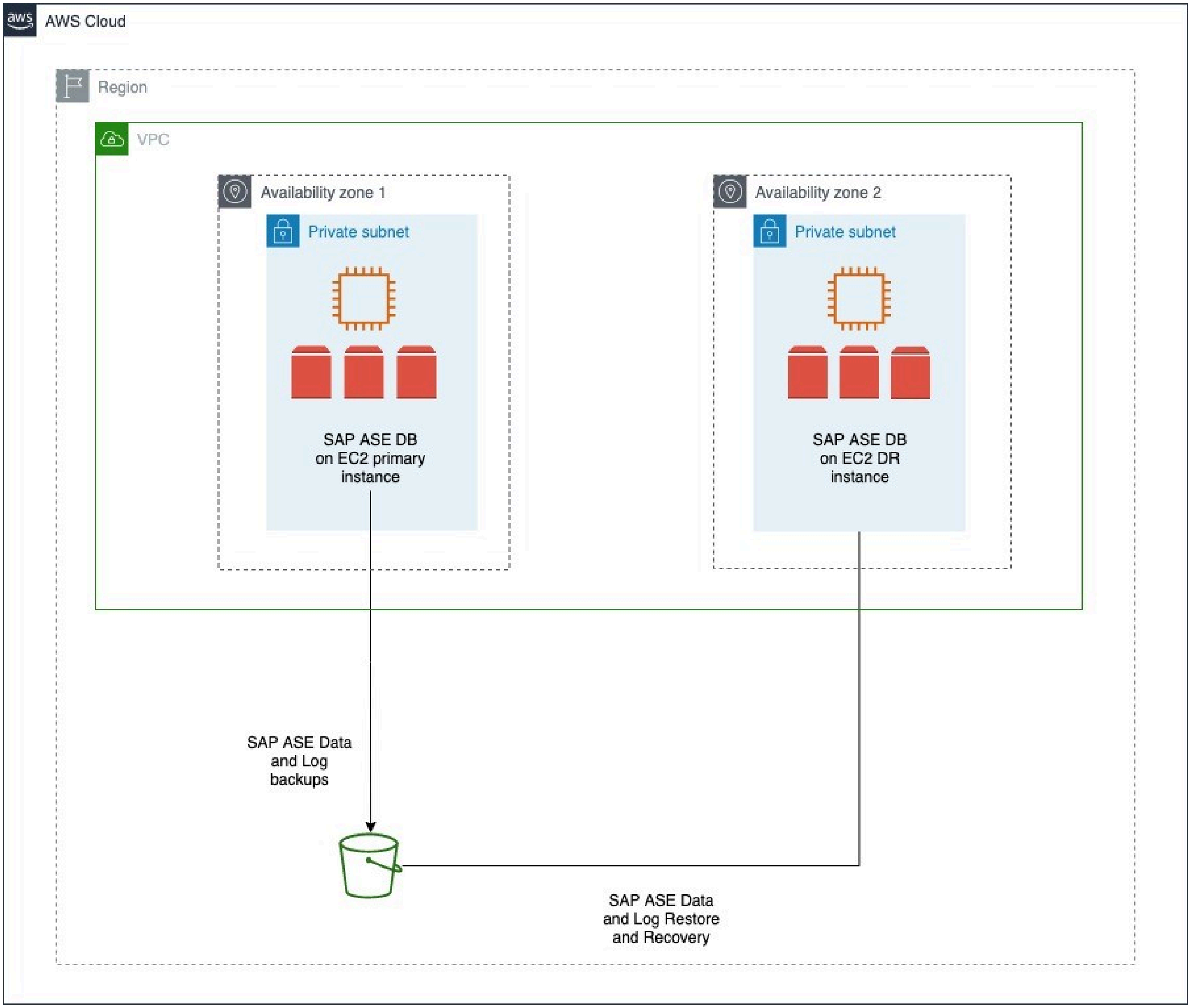
You can store database backups in Amazon S3 and use Amazon S3 Cross-Region Replication (CRR) to replicate your backup in target Region. This method enables automatic, asynchronous copying of objects across Amazon S3 buckets in different AWS Regions. To save costs, you can install and configure SAP ASE database on an Amazon EC2 instance in your disaster recovery Region, and shut the instance. Restart the instance to restore and recover database from the replicated Amazon S3 bucket as needed.

Alternatively, you can use AWS CloudFormation, AWS Cloud Development Kit (AWS CDK) or third-party automation tools to launch an Amazon EC2 instance and to install and configure the SAP ASE database when needed. This helps save costs on Amazon EC2 and Amazon EBS. You must create and test automations before implementation. We recommend performing frequent disaster recovery drills on automations.

The time to recover the database is dependent on the size of the database. Any log files that are not copied over to the disaster recovery Regions are lost and cannot be used for recovery. This option has higher recovery time and point objectives but offers lower costs in comparison to other options. You can use Amazon S3 Replication Time Control to reduce your recovery point objective. For more information, see Using Amazon S3 Replication Time Control.


AWS Cloud
Region 1
VPC
Availability zone
Private subnet
SAP ASE DB on EC2 primary instance
SAP ASE Data and Log backups
Region 2
VPC
Availability zone
Private subnet
SAP ASE DB on EC2 DR instance
SAP ASE Data and Log Restore and Recovery
S3 Cross Region Replication

You can also recover the SAP ASE database backups in the same AWS Region, in case of an Availability Zone failure. Amazon EBS snapshots and Amazon S3 bucket data is automatically replicated within the Region. In the event of an Availability Zone failure, an Amazon EC2 instance can be created in a different Availability Zone of the same Region. It is created from the Amazon EBS snapshots of the source Amazon EC2 instance. The SAP ASE database is restored from the backups in the Amazon S3 bucket. Amazon S3 One Zone-IA is the only exception to automatic replication. For more information, see Amazon S3 Storage Classes.

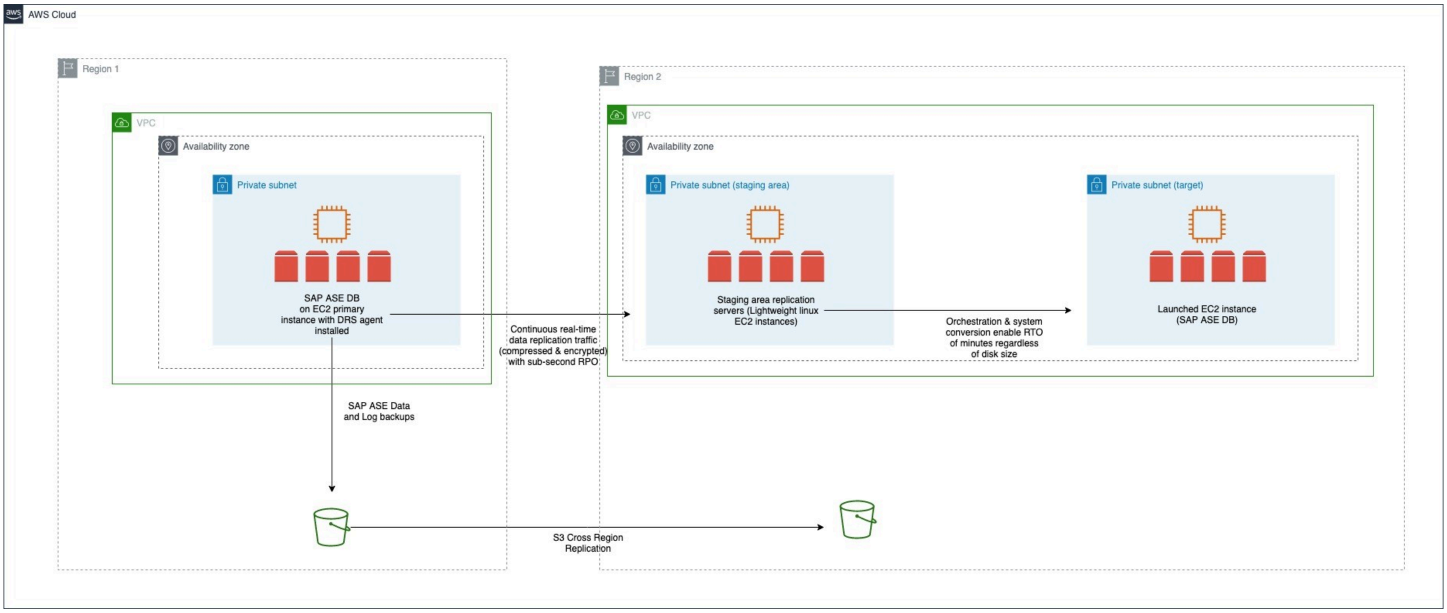

AWS Cloud
Region
VPC
Availability zone 1
Private subnet
SAP ASE DB on EC2 primary instance
Availability zone 2
Private subnet
SAP ASE DB on EC2 DR instance
SAP ASE Data and Log backups
SAP ASE Data and Log Restore and Recovery

Option 3 – disaster recovery using AWS Elastic Disaster Recovery

You can use AWS Elastic Disaster Recovery to replicate source servers from the primary Region to a secondary Region. Elastic Disaster Recovery uses block-level replication, and is not application-aware.

Elastic Disaster Recovery is only used for disaster recovery. You can use Amazon S3 Cross Region Replication for backup.

For more information, see Disaster recovery for SAP workloads on AWS using AWS Elastic Disaster Recovery.


AWS Cloud
Region 1
VPC
Availability zone
Private subnet
SAP ASE DB on EC2 primary instance with DRS agent installed
Continuous real-time data replication traffic (compressed & encrypted) with sub-second RPO
SAP ASE Data and Log backups
S3 Cross Region Replication
Region 2
VPC
Availability zone
Private subnet (staging area)
Staging area replication servers (Lightweight linux EC2 instances)
Orchestration & system conversion enable RTO of minutes regardless of disk size
Private subnet (target)
Launched EC2 instance (SAP ASE DB)

Sizing

Sizing applies to three key areas - compute, network, and storage.

Compute

AWS has certified multiple instance families with different sizes to run SAP workloads. For more details, see Amazon EC2 Instance Types for SAP.

To provision instances based on your requirements, you can use the Right sizing process. This process can help you optimize costs. Although it is ideal to use the right sizing approach when you move your SAP workloads to AWS Cloud, it is an ongoing process. We recommend you to use the latest generation of your selected instance family.

For a greenfield (new) deployment of SAP workloads, you can use the Quick Sizer tool to calculate the compute requirement in SAPS. This helps you to select the closest matching Amazon EC2 instance for a price that is most economical for you. Before completing your selection, ensure that the selected Amazon EC2 instance provides enough Amazon EBS and overall network throughput to meet your application requirements.

For migrations, you can use any of the following data sources to decide the right size of your instance:

- Source system utilization and workload patterns, such as EarlyWatch alert reports.
- Source system specification: CPU, memory, storage size + throughput + IOPS, network.

- Source system SAPS rating.

Network

Network performance is often not explicitly stated as a requirement in SAP sizing. AWS enables you to check the network performance of all Amazon EC2 Instance Types.

Ensure that you have your network components setup to deploy resources related to your SAP workload. If you haven't already setup network components like Amazon VPC, subnets, route tables etc., you can use the, AWS Quick Start Modular and Scalable VPC Architecture to most effectively deploy scalable Amazon VPC architecture in minutes. After setting up your Amazon VPC, you must set up Amazon EC2 instances within the Amazon VPC for your SAP workloads.

Storage

Amazon Elastic Block Store (Amazon EBS) volumes are designed to be highly available and reliable. Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component. Owing to this built-in protection, you can skip configuring `RAID 1` for these volumes.

You must check that the storage required is enough to provide sufficient I/O performance. The new `gp3` volume is ideal for SAP ASE workloads that require smaller volume size. With `gp3`, the storage throughput and IOPS are decoupled from the size and can scale independently.

The io2 volume is well-suited for I/O-intensive database workloads that require sustained IOPS performance or more than 16,000 IOPS. The `io2 Block Express` is another provisioned IOPS SSD volume for workloads that require sub-millisecond latency, sustained IOPS performance, and more than 64,000 IOPS or 1,000 MiB/s of throughput.

Note

`io2 Block Express` is only supported on select Amazon EC2 instance types. For more information, see Provisioned IOPS SSD volumes.

The following table lists the main directories for SAP ASE database.

Usage	Directory

Database instance root files	/sybase/<SID>
Database data files	/sybase/<SID>/sapdata_1 /sybase/<SID>/sapdata_X
Database log files	/sybase<SID>/saplog_1
Database temporary tablespace	/sybase/<SID>/saptmp
Diagnostic tablespace for SAPTOOLS	/sybase/<SID>/sapdiag
Directory for ASE backup	/sybasebackup

Operating system

You can deploy your SAP ASE workload on SLES, SLES for SAP, RHEL for SAP with High Availability and Update Services (RHEL for SAP with HA and US) or RHEL for SAP Solutions.

SLES for SAP and RHEL for SAP with HA and US products are available on AWS Marketplace under an hourly or an annual subscription model.

SLES for SAP

SLES for SAP provides additional benefits, including Extended Service Pack Overlap Support (ESPOS), configuration and tuning packages for SAP applications, and High Availability Extensions (HAE). For details, see the SUSE Linux Enterprise Server for SAP Applications product page to learn more about the benefits of using SLES for SAP. We strongly recommend using SLES for SAP instead of SLES for all your SAP workloads.

If you plan to use Bring Your Own Subscription (BYOS) images provided by SUSE, ensure that you have the registration code required to register your instance with SUSE to access repositories for software updates.

RHEL for SAP

RHEL for SAP with High Availability and Update Services provides access to Red Hat Pacemaker cluster software for High Availability, extended update support, and the libraries that are required to run SAP HANA. For details, see Red Hat Enterprise Linux for SAP offerings on Amazon Web Services FAQ in the Red Hat Knowledgebase.

If you plan to use the BYOS model with RHEL, either through the Red Hat Cloud Access program or other means, ensure that you have access to a RHEL for SAP Solutions subscription. For details, see Overview of Red Hat Enterprise Linux for SAP Solutions subscription in the Red Hat Knowledgebase.

Security and compliance

The following are additional AWS security resources to help you achieve the optimum level of security for your SAP NetWeaver environment on AWS:

- AWS Cloud Security
- CIS AWS Foundations
- AWS Well-Architected Framework

Infrastructure hardening

In some cases, you can further lock down the operating system configuration. For instance, to avoid sharing the credentials of your AWS account with an SAP administrator who needs to log on to an Amazon EC2 instance. Refer to Security in Amazon EC2 and Best Practice 6.2 – Build and protect the operating system to learn more.

You can also use an automated solution provided by AWS – Amazon Inspector.

Encryption

The important aspect of securing your workloads is encrypting your data, both at rest and in transit. For more details, refer to the following resources.

- Amazon EBS encryption
- Data encryption in Amazon EFS
- Data encryption in Amazon S3
- Protect your SAP data at rest and in transit

You can also refer to the following SAP resources.

- *SAP Note 2481596 – SYB: Encrypted data transfer between SAP system and SAP ASE database (requires SAP portal access)*

- SAP Adaptive Server Enterprise – Database Encryption

Security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. Security groups act at the instance level, not the subnet level.

SAP system is often separated into multiple subnets, with the database in a separate subnet to the application servers, and other components, such as a web dispatcher in another subnet, possibly with external access.

If workloads are scaled horizontally, or high availability is necessary, you may choose to include multiple, functionally similar, Amazon EC2 instances in the same security group. In this case, you must add a rule to your security groups.

If Linux is used, some configuration changes may be necessary in the security groups, route tables, and network ACLs. For more information, see Security group rules for different use cases.

Network ACL

A network access control list (ACL) is an optional layer of security for your Amazon VPC that acts as a firewall for controlling traffic in and out of one or more subnets (they're stateless firewalls at the subnet level). You may set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your Amazon VPC.

See Amazon VPC Subnet Zoning Patterns for SAP on AWS to understand the network considerations for SAP workloads.

API call logging

AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you. The recorded information includes the identity of the caller, time of the call, source IP address, request parameters, and response elements returned by the AWS service. With CloudTrail, you can get a history of AWS API calls for your account, including API calls made via AWS Management Console, AWS SDKs, command line tools, and higher-level AWS services (such as, AWS CloudFormation). The AWS API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing.

For more information, see What Is AWS CloudTrail?

Notification on access

You can use Amazon SNS or any third-party application to set up notifications on SSH login to your email address or mobile phone.

Deployment

This section provides information about example deployments.

Standalone deployment

In this example, we set up a sample environment for installation. It includes a public subnet for RDP and SSH access via the internet. We use the AWS Quick Start for Modular and Scalable Amazon VPC Architecture in a single Availability Zone deployment to create the Amazon VPC, subnets, security groups, and IAM roles. You can refer to this example set up but should also follow your own network layout and comply with security standards, such as the following:

- Using a Landing Zone solution like AWS Control Tower.
- Working with a cloud team like Cloud Center of Excellence to use existing standards.

Step 1: Prepare your AWS account

Check the Region where you want to deploy your AWS resources:

- You pick your region for deployment during the planning phase.
- Display the AWS Command Line Interface configuration data:

```
                 $ aws configure list
```

Ensure that the default region listed in the command output is the same as the target region where you want to deploy your AWS resources and install SAP workloads. In this deployment, we provision an Amazon EC2 instance.

Note

In this section, the syntax used for the AWS CLI and Linux commands is specific to the scope of this document. Each command supports many additional options. To learn more, use the `aws help` command.

Step 2: Create a JSON file for Amazon EBS storage

Create a `JSON` file containing the storage requirements for SAP ASE database server volumes. The following is an example `JSON` file with two Amazon EBS volumes for swap and installation directories. You can add more volumes as per your storage design.

```
[
    {
        "DeviceName": "/dev/nvme2n1",
        "Ebs": {
        "VolumeSize": 32,
        "VolumeType": "gp3",
        "DeleteOnTermination": true
        }
    },
    {
        "DeviceName": "/dev/nvme3n1",
        "Ebs": {
        "VolumeSize": 50,
        "VolumeType": "gp3",
        "DeleteOnTermination": true
    }
  }
]
```

Note

In the preceding example, the device name `/dev/nvme2n1` is for Nitro based hypervisors. It differs for non-Nitro based hypervisors. For more information, see Storage configuration.

Step 3: Launch the Amazon EC2 instance

Launch the Amazon EC2 instance for the SAP ASE database installation in your target AWS Region, using the information gathered in Step 1. You must create the required storage volumes and attach them to the Amazon EC2 instance for the SAP installation, based on the `JSON` file you created in the Amazon EBS storage (Step 2).

```
$ aws ec2 run-instances \
--image-id <AMI-ID> \
--count <number-of-EC2-instances> \
--instance-type <instance-typ> \
--key-name=<name-of-key-pair> \
--security-group-ids <security-group-ID> \
--subnet-id <subnet-ID> \
--block-device-mappings file://<PATH>\<file>.json \
--region <region-ID>
```

Use this command in a single line format, as shown in the following example.

```
aws ec2 run-instances --image-id ami-xxxxxxxxxxxxxxx --count 1 --instance-type m5.large
 --key-name=my_key --security-group-ids sg-xxxxxxxx --subnet-id subnet-xxxxxx --block-
device-mappings file://<PATH>\<file>.json
```

In this example, *m5.large* is the value for the `instance-type` parameter. You must select an Amazon EC2 instance type based on your business requirements.

You can also launch Amazon EC2 instances using the AWS Management Console. For more information, see Launch an instance.

Step 4: Prepare the Linux Operating System

Before starting the installation, you need to perform Linux specific prerequisite tasks. For more information, refer to the following SAP Notes (requires SAP portal access).

- SAP Note 1554717 – SYB: Planning information for SAP on ASE
- SAP Note 1748888 – SYB: Inst.Systems Based on NW 7.3 and Higher: SAP ASE

Step 5: Prepare each Amazon EC2 instance for SAP ASE installation

Download the SAP installation media as per the SAP installation guide, for the version of SAP NetWeaver you want to install on your Amazon EBS volumes. Locate your installation guide on the Guide Finder for SAP NetWeaver and ABAP Platform. You can store the SAP installation media using Amazon EFS or an Amazon S3 bucket for later reuse.

If you choose to install the SAP ASE database with high availability deployment across two Availability Zones, repeat the preceding steps for SAP ASE database standby high availability instance in the second Availability Zone.

If you choose to install SAP ASE database with high availability and disaster recovery deployment across two AWS Regions, repeat the preceding steps in the second AWS Region in which you want to run the ASE database standby disaster recovery instance.

Step 6: Installing SAP ASE on Amazon EC2 instances

You are now ready to install the SAP ASE software on your Amazon EC2 instances. For more information, see the SAP ASE Database Software Installation section of your SAP NetWeaver installation guide. Locate your installation guide on the Guide Finder for SAP NetWeaver and ABAP Platform.

The following is a non-exhaustive list of post-installation tasks for your SAP ASE database.

- Updating to the most recent patch available
- Installation of additional components
- Configure the SAP ASE backup

For more information, see the Operations section.

High availability disaster recovery deployment

Create an additional Amazon EC2 instance and perform the installation in a secondary Availability Zone. The steps for creating a high availability or disaster recovery instance in a secondary Availability Zone are the same as described in Standalone deployment. You can simplify this step by using the following methods.

- If you have built any automation using AWS CloudFormation or other tools to create the primary Amazon EC2 instance and install database software, you can use the same automation to build the HA instance.
- You can create an Amazon Machine Image of the primary Amazon EC2 instance and launch another instance in the secondary Availability Zone.

The configuration of high availability or disaster recovery depends on the tools you use. See the next sections for more details.

> **Note**
>
> You must configure cross-regional Amazon VPC peering or Transit Gateway to enable SAP ASE asynchronous replication between two Regions.

Operations

Tagging AWS resources

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value, both defined by you. Adding tags to various AWS resources will make managing SAP environments more efficient, and help you search for resources quickly. Many Amazon EC2 API calls can be used in conjunction with a special tag filter. For more information, see Tagging AWS resources. The following are some examples of how you can use tags for your operational needs.

Tag name	Tag value
Name	SAP server’s virtual (host) name
Environment	SAP server’s landscape role; for example: SBX, DEV, QAT, STG, PRD.
Application	SAP solution or product; for example: ECC, CRM, BW, PI, SCM, SRM, EP
Owner	SAP point of contact
Service level	Known uptime and downtime schedule

After tagging your resources, you can apply specific security restrictions, such as access control (as seen in the following example policy), based on tag values.

```
            {
                "Version": "2012-10-17",
                "Statement": [
                    {
                    "Sid": "LaunchEC2Instances",
                     "Effect": "Allow",
                    "Action": [
                        "ec2:Describe*",
                         "ec2:RunInstances"
                    ],
                    "Resource": [
                        "*"
                    ]
                    },
                    {
                    "Sid": "AllowActionsIfYouAreTheOwner",
                    "Effect": "Allow",
                    "Action": [
                        "ec2:StopInstances",
                        "ec2:StartInstances",
                        "ec2:RebootInstances",
                         "ec2:TerminateInstances"
                    ],
                    "Condition": {
                        "StringEquals": {
                        "ec2:ResourceTag/PrincipalId": "${aws:userid}"
                        }
                    },
                    "Resource": [
                        "*"
                    ]
                    }
                ]
            }
```

IAM only allows specific permissions based on the tag value. In this scenario, the current ID must match the tag value to enable permissions for the user. For more information, see Tag your Amazon EC2 resources.

Monitoring

AWS provides multiple native services to monitor and manage your SAP environment. CloudWatch and CloudTrail can be used to monitor your underlying infrastructure and APIs. CloudWatch provides ready-to-use KPIs for CPU, disk utilization, and enables you to create custom metrics for KPIs that you want to monitor. CloudTrail allows you to log the API calls made to your AWS infrastructure components.

Operating system maintenance

In general, operating system maintenance across large estates of Amazon EC2 instances can be managed by using:

- Third-party products, such as those available on AWS Marketplace.
- AWS Systems Manager

The following are some key operating system maintenance tasks.

Patching

You can follow SAP recommended patching process to update your landscape on AWS. With AWS Systems Manager Patch Manager, you can roll out OS patches according to your corporate policies. It has multiple benefits:

- Scheduling based on tags
- Defining patch baselines
- Auto-approving patches with lists of approved and rejected patches

AWS Systems Patch Manager integrates with IAM, CloudTrail, and CloudWatch Events to provide a secure patching experience that includes event notifications and the ability to audit usage. For details about the process, see How Patch Manager operations work. Third-party products are available on AWS Marketplace.

Maintenance Windows

AWS Systems Manager Maintenance Windows lets you define a schedule to perform potentially disruptive actions on your instances, such as patching an operating system, updating drivers, installing software or patches.

Administrator access

For administrative purposes, you can access the backend of your SAP systems via SSH or AWS Systems Manager Session Manager.

Automation

AWS Systems Manager Automation simplifies common maintenance and deployment tasks of Amazon EC2 instances and other AWS resources. For more information, see AWS Systems Manager Automation.

Automation using Infrastructure-as-Code with AWS CloudFormation

We recommend following the principle of Infrastructure-as-Code (IaC) for automating and maintaining your workloads on AWS. AWS CloudFormation provides a common language for you to describe and provision all the infrastructure resources in your cloud environment in a repeatable and automated manner.

Cost optimization

We recommend cost optimization as an ongoing process. There are many AWS services that help with budgeting, cost control and optimization. For more details, see Cost Optimization Pillar - AWS Well-Architected Framework and Cost Optimization Pillar -SAP Lens.

Compute & storage

Compute

Amazon EBS volumes are exposed as NVMe block devices on Instances built on the Nitro System. When changing Amazon EC2 instance types from a previous generation to a Nitro generation, NVMe device IDs associated with the volume can change. To avoid mount errors during change of instance type or instance reboots, you need to create a label for your file systems and mount it by the label, *and not* the NVMe IDs. For more details, see support article.

Aside from operating system maintenance, you should consider maintenance for your Amazon EC2 instances. It can be driven by using Creating your own runbooks. The following are some examples.

- Use `AWS-StopEC2InstanceWithApproval` to request one or more IAM users approve the instance stop action. After the approval is received, runbook stops the instance.

- Use `AWS-StopEC2Instance` to automatically stop instances on a schedule, using CloudWatch Events or a Maintenance Window task. For example, you can configure an Automation workflow to stop instances every Friday evening and restart on Monday mornings. Note that this automation will only stop and start the Amazon EC2 instance. You must create additional document to gracefully stop and start SAP applications and database and then use the AWS Systems Manager to run such automations.
- Use `AWS-UpdateCloudFormationStackWithApproval` to update resources that were deployed using AWS CloudFormation template. The update applies a new template. You can configure the Automation to request approval by one or more IAM users before the update begins.

You can also use AWS Instance Scheduler to configure custom start and stop schedules for Amazon EC2 and Amazon RDS instances.

Storage

The following are the storage services used across this guide.

- Amazon EBS provides persistent storage for SAP applications and database. Amazon EBS volumes can be resized and even have the volume type changed without disrupting the applications. For more details, see Amazon EBS Elastic Volumes. After modifying the Amazon EBS volume, you need to extend the file system to match the extended volume size. For more details, see Extend a Linux file system after resizing a volume.
- Amazon EFS does not require you to explicitly provision storage, you pay only for your usage. It is built to scale on demand, without disrupting applications, growing and shrinking automatically as you add and remove files. This ensures that your applications have the required storage.
- Amazon S3 also does not require you to explicitly provision storage, you pay only for your usage. You can use Object lifecycle management to set rules that define when objects are transitioned or archived to colder storage (Amazon S3 IA or S3 Glacier) and when they expire. For more information, see Managing your storage lifecycle.

Backup & restore

Snapshots and AMIs

A common approach for backing up your SAP NetWeaver application servers is using snapshots and AMIs.

The SAP application data is stored on Amazon EBS volumes attached to the SAP NetWeaver application servers. You can back up the data on these volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups of Amazon EBS volumes, which means that only the blocks on the device that have changed after your most recent snapshot are saved. For more information, see Create Amazon EBS snapshots.

An Amazon Machine Image (AMI) provides the information required to launch an instance along with a block device mapping of all Amazon EBS volumes attached to it.

Amazon EC2 powers down the instance before creating the AMI to ensure that everything on the instance is stopped and in a consistent state during the creation process. If you're confident that your instance is in a consistent state appropriate for AMI creation, you can check the *No Reboot* option.

You can use AWS Backup to centrally configure backup policies and monitor backup activity for these snapshots. Once you have completed the SAP installation and post-installation steps, create an image of the instance.

```
aws ec2 create-image --instance-id i-1234567890abcdef0 --name "My server" --description
 "An AMI for my server"
```

AWS provides a very simple and quick way to copy an SAP system. You can use the AWS Console Home or the AWS CLI to create a new AMI of an existing SAP system. You can then launch exact copies of the original system from the new AMI. For more details, see Amazon Machine Images (AMI).

Backup to Amazon S3

You can perform traditional file-based backup to Amazon S3 from your Amazon EBS volumes. One way to take backup is to use AWS CLI and initiate it by using AWS Systems Manager Run command, so that you can centrally manage the backups.

Backup with third-party products

Many third-part products for AWS services are certified by SAP. For more information, see AWS SAP Competency Partners.

Amazon EFS backup

Using AWS Backup, you can centrally configure backup policies and monitor backup activity for AWS resources, including Amazon EFS file systems.

Alternatively, you can perform a file-level backup of your Amazon EFS file system to Amazon S3. You can do this by running a file-level copy to Amazon S3 from any Amazon EC2 instance running in the same region. This can be automated and scheduled using AWS Systems Manager Run Command in combination with CloudWatch Events.

Backup and restore for ASE database

You must to regularly backup your operating system and database to recover them in case of any failure. AWS Cloud provides various services and tools that you can use to backup your SAP ASE database.

Storage snapshots

You can backup your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only blocks on the device that have changed after your most recent snapshot are saved. Snapshots of Amazon EBS volumes can be created for backup of SAP ASE database file systems.

See How to use snapshots to create an automated recovery procedure for SAP ASE databases to learn more.

SAP ASE database backups

You can configure your SAP ASE database to store backups on Amazon EFS or local Amazon EBS volumes. You must configure regular backups for Amazon EFS. For more information, see Backing up your Amazon EFS file systems. You can reduce costs by enabling Amazon EFS storage classes to retain cold backups in infrequent access. For more information, see Amazon EFS Infrequent Access.

You can also configure backups to be store on Amazon EFS volumes and to be regularly uploaded to Amazon S3. Use `DBACOCKPIT` to schedule backup frequency. You can also use AWS Systems Manager Maintenance Windows to schedule backup frequency.

Amazon SNS enables you to setup push notifications for success or failure. Once backups are stored in Amazon S3, you can use lifecycle policies to define data retention timeline. For more information, see Managing your storage lifecycle.

You can improve Amazon S3 data upload performance with Gateway endpoints and AWS CLI. For more information, see Gateway endpoints for Amazon S3 and AWS CLI S3 Configuration.

Review the following SAP Notes (portal access required) for more details.

- SAP Note 1585981 - SYB: Ensuring Recoverability for SAP ASE
- SAP Note 1887068 - SYB: Using external backup and restore with SAP ASE
- SAP Note 1588316 - SYB: Configure automatic database and log backups
- SAP Note 1618817 - SYB: How to restore an SAP ASE database server (UNIX)

To use third-party tools to backup your SAP ASE database, see AWS Storage Competency Partners.

Disaster recovery

See Disaster recovery deployment to learn about disaster recovery for your SAP ASE database.

Perform a DNS change

In case of manual failover, you may install SAP application servers using a virtual hostname and perform a DNS change to direct the SAP application servers to the new primary database server. For a DNS resolution in AWS, you can use any of the following options.

- Amazon Route 53 enables you to create a private hosted zone for your environment and an A record for the virtual hostname used for SAP ASE database. Initially, this A record is mapped to the IP address of the primary SAP ASE database instance.
- You can maintain your own DNS server on-premise or on your Amazon EC2 instances. You can create an A record there for your virtual hostname used for SAP ASE database. Initially, this A record is mapped to the IP address of the primary SAP ASE database instance.
- With the AWS Directory Service, you can create an A record for the virtual hostname used for SAP ASE database.

With any of the previously mentioned options, you can change the A record to a private IP address of the primary database instance in case of a failover. This DNS change can also be automated using AWS services and scripts.

Resources

SAP on AWS customers have the flexibility to deploy SAP ASE database on the scalable, on-demand Amazon EC2 platform in a highly available manner. They don’t have to invest in costly capital expenditures for the underlying infrastructure. By combining the AWS platform flexibility and SAP installation techniques, our customers greatly improve the availability of their deployments. For more details, see SAP on AWS Case Studies.

Support

AWS offers three levels of support. AWS Business Support provides resources and technical support for customers running SAP workloads on AWS. AWS Enterprise Support and AWS Enterprise On-Ramp Support offers support to customers running mission critical SAP production workloads on AWS.

To learn more about this, see SAP Note 1656250 – SAP on AWS: Support prerequisites (requires SAP portal access).

SAP NetWeaver on AWS Automation

AWS Systems Manager is a collection of capabilities that help you manage your applications and infrastructure running in AWS Cloud. Systems Manager simplifies application and resource management, shortens the time to detect and resolve operational problems, and helps you manage your AWS resources securely at scale.

This chapter contains information about how to use Systems Manager to automate management of your SAP applications.

Automation prerequisites

Because SAP automation in AWS Cloud relies on Systems Manager, you must satisfy the Systems Manager prerequisites. In addition, there are prerequisites specified in this chapter for specific tasks, such as SAP installation and operating system patching. Those prerequisites are listed in their respective sections.

Before you begin, verify the following prerequisites, which apply to all of the automation tasks described in this chapter:

- You must have the latest SSM agent installed on your Amazon EC2 instances. For more information, see Manually installing SSM Agent on EC2 instances for Linux in the *AWS Systems Manager User Guide*.
- You must satisfy the prerequisites for Systems Manager. For more information, see Systems Manager prerequisites in the *AWS Systems Manager User Guide*.

Automated SAP installation

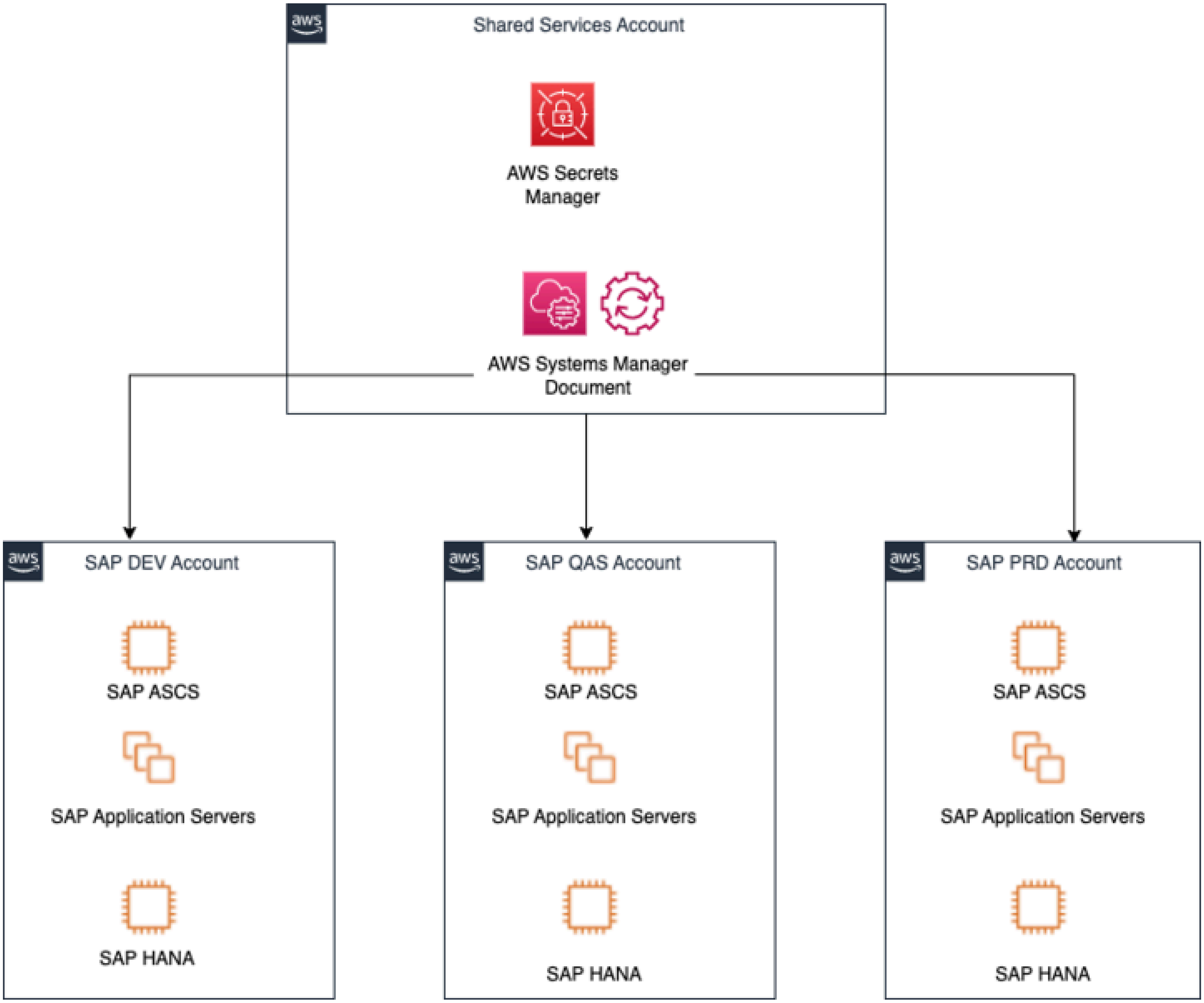
Deploying an SAP system requires significant effort in building an infrastructure that conforms to SAP specifications. Installation, operating system configuration, and configuration of parameters based on the type of SAP workload must be repeated for the development, quality, and production landscape. You can automate this installation and configuration using AWS Systems Manager. Automating the installation and configuration of your SAP landscape helps your team stay compliant with auditable policies related to configuration as code. In addition, it turns the SAP installation into an easily repeatable process, which makes the quality of the outcome easier to improve because you can simulate it and run it multiple times using the same source of information.

The solution described here uses Systems Manager documents to install a distributed SAP landscape that contains the following:

- ABAP SAP Central Services (ASCS)
- Database instance
- Primary Application Server (PAS)
- Additional Application Servers (AAS)

Automated SAP installation architecture

The example architecture shown in the diagram below uses a centralized AWS account that stores the AWS Systems Manager document (SSM document). The document is shared with AWS accounts that host Amazon EC2 instances running SAP HANA workloads.



You can use multiple AWS accounts and AWS organizations to arrange the accounts into a hierarchy and group them into organizational units. These organizational units can be used for things such as consolidated billing, workload isolation, and administrative isolation. You can create separate AWS accounts for development, testing, staging, and production on a per-application basis as part of an organization. For more information, see the https://docs.aws.amazon.com/organizations/latest/userguide/orgs_introduction.html *AWS Organizations">User Guide*.

Systems Manager automation provides multi-account and multi-AWS Region support that allows you to execute your own automation documents across multiple accounts from a central AWS account. You can centralize the SSM documents into a Shared Services account or use an automation account. The automation account can be the AWS account that runs SAP workloads or a dedicated account that only runs SSM documents. Using a centralized AWS for automation reduces administration overhead by maintaining the SSM document and its dependencies in a single account. For more information about Shared Services, see Infrastructure OU - Shared Services account in the *AWS Security Reference Architecture*.

In order for Systems Manager to trigger automation documents from a centralized AWS account to the connected accounts, IAM permissions are required in the automation and child accounts. For more information, see Running automations in multiple AWS Regions and accounts in the *AWS Systems Manager User Guide*.

You can share SSM documents privately or publicly with accounts in the same Region. To privately share a document, modify the document permissions and allow specific individuals to access it based on their AWS account ID. For more information, see Sharing SSM documents in the *AWS Systems Manager User Guide*.

Components

The installation automation workflow includes automation runbooks and SSM command documents.

Automation runbook

An automation runbook defines the actions that Systems Manager performs on your managed instances and other AWS resources. A runbook contains one or more steps that run in sequential order. For more information, see the following documentation:

- What is an automation? in the *AWS Systems Manager User Guide*
- Systems Manager Automation runbook reference

SSM command document

If a task must be repeated multiple times on multiple hosts, you can create it as an SSM command document. These documents are usable across multiple runbooks. For more information, see Systems Manager Command document plugin reference in the *AWS Systems Manager User Guide*.

You can make the SSM command document as granular as you need, based on factors such as:

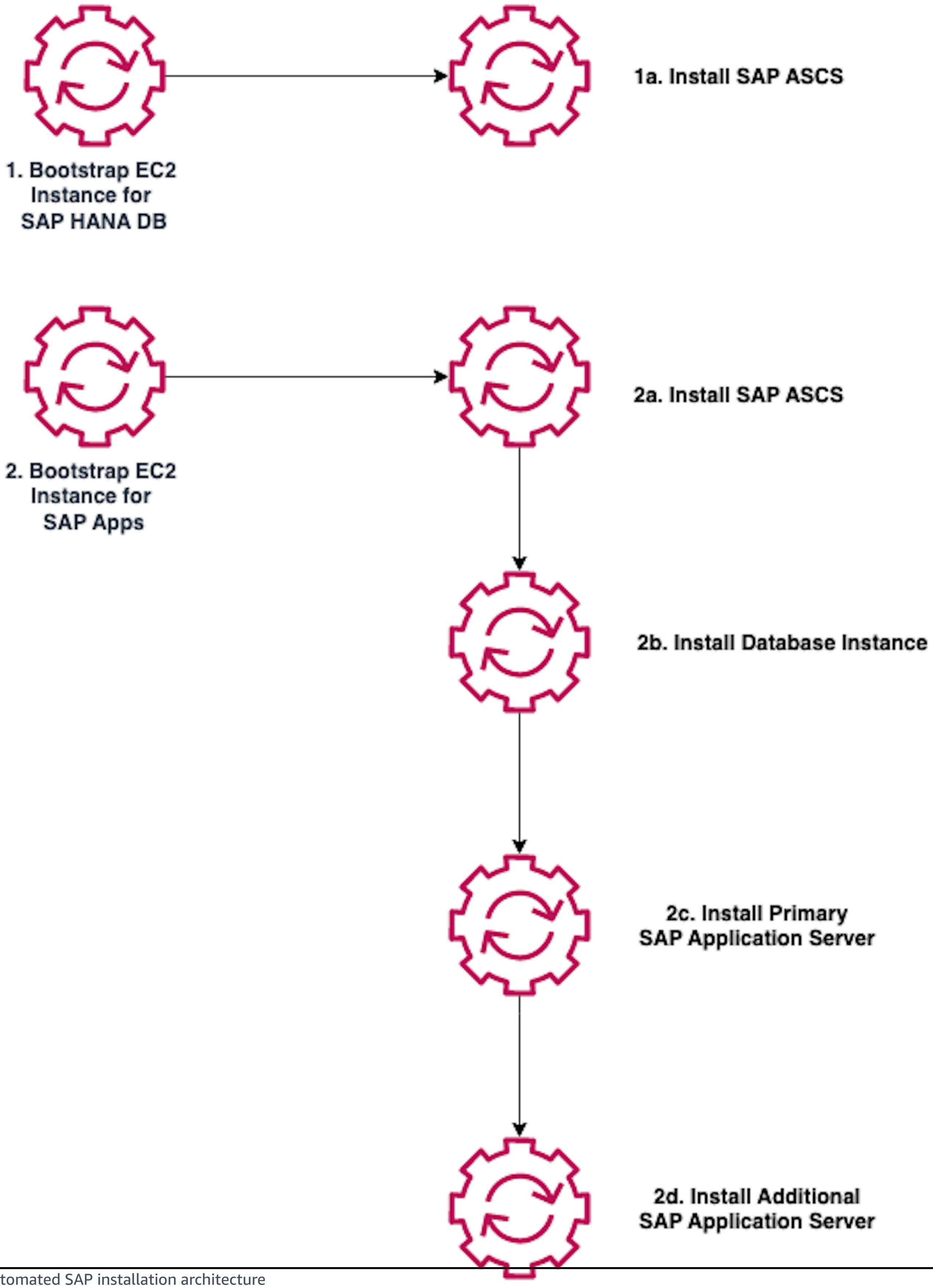
- Segregation of duties
- Types of SAP systems that are being deployed

- Complexity of SAP systems that are being deployed
- Security

Workflow

As an example, each runbook can be made up of several SSM documents that perform a specific configuration. The following runbooks can be used, which are illustrated in the diagram below.

- Bootstrap Amazon EC2 instances for SAP HANA database
- Bootstrap Amazon EC2 instances for SAP application servers
- Install SAP HANA database
- Install ABAP SAP Central Services (ASCS)
- Install a database instance
- Install a primary application server
- Install an additional application server


1. Bootstrap EC2 Instance for SAP HANA DB
1a. Install SAP ASCS
2. Bootstrap EC2 Instance for SAP Apps
2a. Install SAP ASCS
2b. Install Database Instance
2c. Install Primary SAP Application Server
2d. Install Additional SAP Application Server

Automated SAP NetWeaver on AWS installation prerequisites

In addition to the prerequisites described in the Automation prerequisites section of this guide, verify the following prerequisites that are specific to automated SAP installation:

- You must have an existing infrastructure deployed.

  The example described in this guide uses a SAP HANA database, an SAP Central Services (ASCS) instance, and a database instance. The *AWS for SAP* blog has a Terraform your SAP Infrastructure on AWS example.
- SAP media files must be available.

  You must provide the SAP installation media files, which are obtained from SAP, in an Amazon S3 bucket. For more information, see Make SAP application software available for AWS Launch Wizard for SAP to deploy SAP in the *AWS Launch Wizard User Guide*. If you use the sample code provided in this guide, the media files are copied to local Amazon Elastic Block Store volumes.

SAP Notes

Read the following SAP Note:

- SAP Note: 2230669 - System Provisioning Using a Parameter Input File

Additional references

Before you begin, you can also familiarize yourself with how SAP works on AWS by reading the following documentation:

- SAP on AWS Planning in the *General SAP Guides*
- Amazon EC2 instance types for SAP on AWS in the *General SAP Guides*
- SAP NetWeaver Environment Setup for Linux on AWS in the *SAP NetWeaver Guides*

Configuring automated SAP installation

The sections below contain detailed instructions on how to configure automated SAP NetWeaver on AWS installation.

Customize the Systems Manager document

This section shows you how to customize the AWS Systems Manager document (SSM document) for the automated SAP installation. For more information about SSM documents, see AWS Systems Manager Documents in the *AWS Systems Manager User Guide*.

This section details the content that goes into the SSM document. For information about how to create the document, see Create an SSM document (console) in the *AWS Systems Manager User Guide*.

As you create your SSM document, we recommend you do the following:

- Use schema version 2.2. For more information, see SSM document schema features and examples in the *AWS Systems Manager User Guide*.
- Use Parameter Store to easily reference parameters that you use often. For more information, see AWS Systems Manager Parameter Store in the *AWS Systems Manager User Guide*.

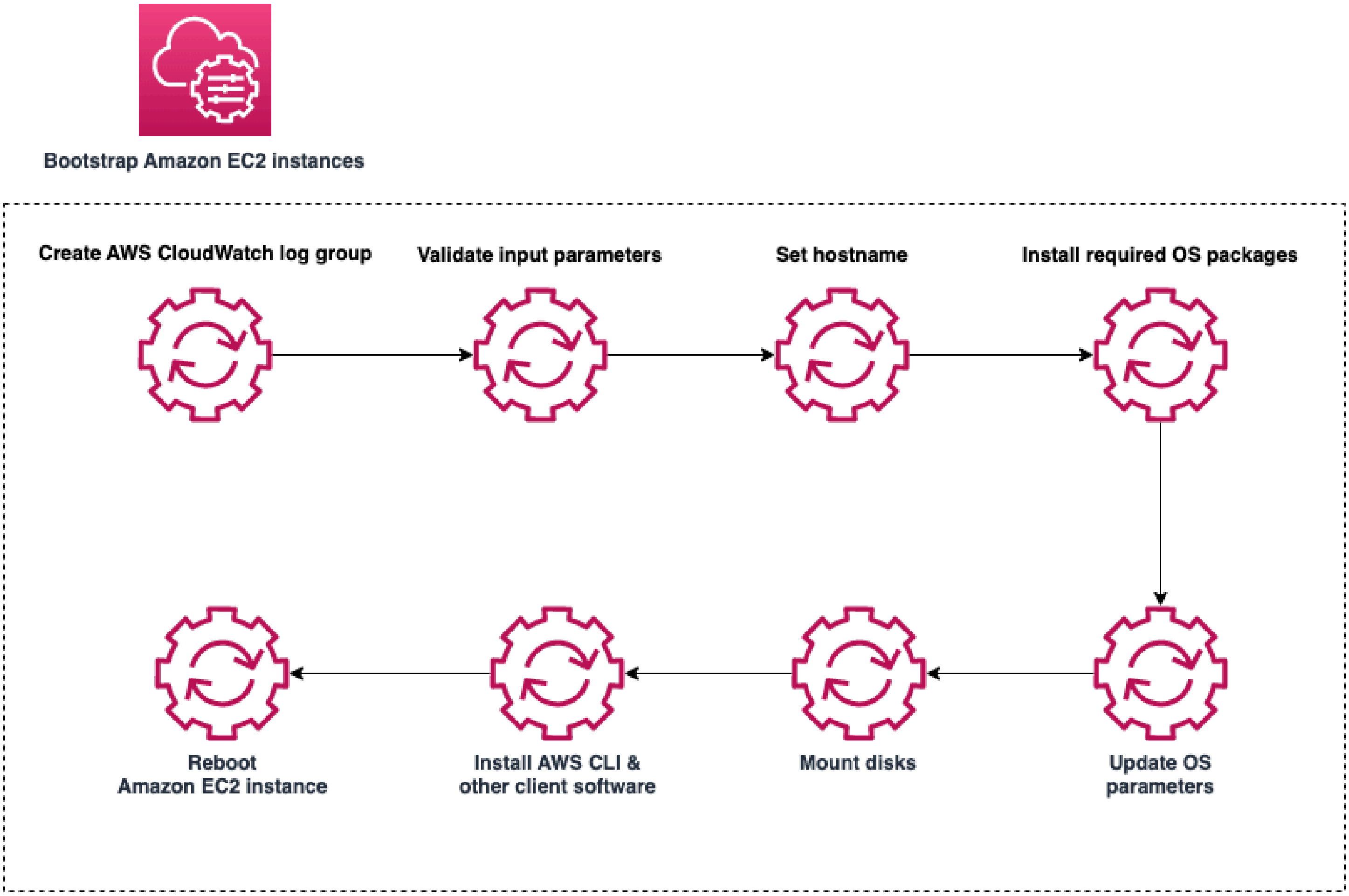
> **Tip**
>
> You can find sample SSM documents and parameter files in the aws-samples/terraform-aws-sap-netweaver-on-hana GitHub repository.

Bootstrap Amazon EC2 instances

Bootstrapping in Amazon EC2 consists of adding commands or scripts to the user data section of the instance. These commands and scripts can be executed when the instance starts. This simplifies configuration tasks. For more information, see Run commands on your Linux instance at launch in the *Amazon Elastic Compute Cloud User Guide for Linux Instances*.

For SAP installation, bootstrapping includes several tasks, such as setting the hostname, installing operating system packages, setting operating system parameters, installing AWS Data Provider for SAP, installing agents for monitoring, logging, and alerting, and mounting disks for the SAP HANA database instance and SAP application servers.

The image below shows the steps required for the bootstrap instance SSM document.


Bootstrap Amazon EC2 instances
Create AWS CloudWatch log group
Validate input parameters
Set hostname
Install required OS packages
Update OS parameters
Mount disks
Install AWS CLI & other client software
Reboot Amazon EC2 instance

The SSM document accepts required and optional parameters. The code below is an example parameter section for bootstrapping an SAP HANA database instance or any SAP NetWeaver application server instance:

```
parameters:
    AutomationAssumeRole:
        type: String
        description: "(Optional) The ARN of the role that allows Automation to perform
 the actions on your behalf. "
        default: ''
    InstanceId:
        type: String
        description: "(Required) The instance ids to bootstrap before SAP HANA
 installation"
        default: ''
    HostnameTagKey:
        type: String
        description: "(Required) The tag key where the hostname is stored"
        default: 'Hostname'
    DnsPrivateZoneName:
        type: String
        description: "(Optional) DNS Zone name to specify FQDN in hosts"
        default: 'sapteam.net'
    EfsFileSystemId:
        type: String
        description: (Required) The EFS file system id for /sapmnt folder
        default: ‘fs-7df7edae’
    MasterPassword:
        type: String
        description: '(Required) SAP NetWeaver Master Password'
        default: ''
    IniFile:
        type: String
        description: '(Required) Path to INI file'
        default: '/sapmnt/software/sapinstall.params'
    CloudWatchLogGroupName:
        type: String
        description: "(Required) Cloud Watch log group for the log output"
        default: '/customer/SAP/dev-setup-logs'
```

The next section of the SSM document is the `mainSteps` section.

A composite SSM document is a custom document that performs a series of actions by running one or more secondary SSM documents. Composite documents promote infrastructure as code by allowing you to create a standard set of SSM documents for common tasks, such as bootstrapping software or domain-joining instances. For example, you can create a composite document with secondary SSM documents for each bootstrap item, as listed below:

- Setting the hostname
- Installing operating system packages for SAP HANA
- Setting the operating system parameters for SAP HANA
- Mounting disks for SAP HANA
- Installing the AWS Data Provider agent for SAP

Composite and secondary documents can be stored in Systems Manager, private and public GitHub repositories, or Amazon S3. They can be created in JSON or YAML. For more information, see Creating composite documents in the *AWS Systems Manager User Guide*.

The code below shows the mainSteps section of the SSM document with the composite and secondary documents:

```
mainSteps:
- name: Prepare_logs
  action: aws:runCommand
  inputs:
    DocumentName: d4h-prepare-sap-installation-logs
    InstanceIds:
    - '{{ InstanceId }}'
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: '{{ CloudWatchLogGroupName }}'
      CloudWatchOutputEnabled: True
- name: Set_hostname
  action: aws:runCommand
  inputs:
    DocumentName: d4h-set-hostname
    InstanceIds:
    - '{{ InstanceId }}'
    Parameters:
      PrivateZone: '{{ DnsPrivateZoneName }}'
      Hostname: '{{ Get_hostname.Hostname }}'
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: '{{ CloudWatchLogGroupName }}'
      CloudWatchOutputEnabled: True
- name: Install_Packages
  action: aws:runCommand
  inputs:
    DocumentName: d4h-install-sap-packages
    InstanceIds:
    - '{{ InstanceId }}'
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: '{{ CloudWatchLogGroupName }}'
      CloudWatchOutputEnabled: True
- name: Set_OS_Parameters
  action: aws:runCommand
  inputs:
    DocumentName: d4h-set-sap-hana-parameters
    InstanceIds:
    - '{{ InstanceId }}'
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: '{{ CloudWatchLogGroupName }}'
      CloudWatchOutputEnabled: True
```

```
- name: Mount_Disks
  action: aws:runCommand
  inputs:
    DocumentName: d4h-mount-hana-disks
    InstanceIds:
    - '{{ InstanceId }}'
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: '{{ CloudWatchLogGroupName }}'
      CloudWatchOutputEnabled: True
- name: Install_Aws_Sap_Data_Provider
  action: aws:runCommand
  isCritical: false
  inputs:
    DocumentName: d4h-install-sap-aws-data-provider
    InstanceIds:
    - '{{ InstanceId }}'
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: '{{ CloudWatchLogGroupName }}'
      CloudWatchOutputEnabled: True
```

Install the SAP HANA database

After you bootstrap the Amazon EC2 instances, you must install the SAP HANA database. For this installation, you can store the SAP HANA master password in the SSM document Parameter Store or use it as an input to the SSM document and reference it in the `passfile.xml` file.

The code below is an example SSM document for an SAP HANA installation:

```
mainSteps:
- action: "aws:runShellScript"
  name: "Run_installer"
  inputs:
    runCommand:
    - #!/bin/bash
    - HANA_MEDIA=`find /software/hana -name "DATA_UNITS"`
    - if [ -z "$HANA_MEDIA" ]
    - then
    -   echo "Could not find the DATA_UNITS folder in /software/hana. Check if
 everything was downloaded successfully. Exiting..." | tee -a $SSM_LOG_FILE
    -   exit 1
    - fi
    - PASSFILE=$HANA_MEDIA/../passfile.xml
    - chmod +x $HANA_MEDIA/HDB_LCM_LINUX_X86_64/hdblcm
    - HOSTNAME=`(hostname)`
    - INSTANCE=`(instancenumber)`
    - SID=`echo "{{sid}}" | tr a-z A-Z`
    - echo "Executing installation from $HANA_MEDIA/HDB_LCM_LINUX_X86_64/hdblcm for SID
 $SID, instance $INSTANCE, hostname $HOSTNAME..."
    - cat $PASSFILE | $HANA_MEDIA/HDB_LCM_LINUX_X86_64/hdblcm --action=install --
components=client,server --batch --autostart=1 -sid=$SID  --hostname=$HOSTNAME --
number=$INSTANCE  --read_password_from_stdin=xml | tee -a $SSM_LOG_FILE
    - echo "`date` Installation finished. Please check logs..." | tee -a $SSM_LOG_FILE
    - rm $INIFILE
```

Install SAP

Installing SAP includes ABAP SAP Central Services (ASCS), the database instance, and the primary and additional application server installation.

First, you create a parameter file with the required parameters. Refer to the SAP installation guide for the parameters that are specific to your installation. The code below is an example parameter file:

```
mainSteps:
- action: "aws:runShellScript"
  name: "Prepare_sapinstall_ini"
  inputs:
    runCommand:
    - #!/bin/bash
    - SAPINSTALL_INI_FILE={{ IniFile }}
    - SID=`echo "{{Sid}}" | tr a-z A-Z`
    - SAPSYSUID=`sapsysuid`
    - SIDADMUID=`sidadmuid`
    - SWTARGET=/sapmnt/software/
    - DOMAINNAME={{ DnsPrivateZoneName }}
    - HOSTNAME=`hostname`
    - FQDN=${LHOSTNAME}.${DOMAINNAME}
    - sed -i "s|default_scsVirtualHostname|${HOSTNAME}|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_scsInstanceNumber|00|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_ssmpass|{{ MasterPassword }}|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_sid|${SID}|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_fqdn|${DOMAINNAME}|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_sapsysGID|${SAPSYSUID}|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_AdmUID|${SIDADMUID}|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_downloadBasket|${SWTARGET}|g" ${SAPINSTALL_INI_FILE}
    - echo '`date` Prepared the Ini File:...' | tee -a $SSM_LOG_FILE
```

The next step is to start the installation using the SAP silent, or unattended, installation mode, referring to the parameter file as in the example code below:

```
mainSteps:
- action: "aws:runShellScript"
  name: "Execute_installation"
  inputs:
    runCommand:
    - #!/bin/bash
    - echo '`date` Starting the Installation process...' | tee -a $
    - SYSTEMNUMBER=`systemnumber`
    - SAPAliasName=`hostname`
    - SWPMFILE=`find /sapmnt/software/SWPM-SUM/ -name SWPM*SAR`
    - chmod 775 /sapmnt/software/utils/sapcar
    - /sapmnt/software/utils/sapcar -xvf $SWPMFILE -R /sapmnt/software/SWPM
    - chmod 755 /sapmnt/software/SWPM/sapinst
    - cd /sapmnt/software/SWPM
    - ./sapinst SAPINST_INPUT_PARAMETERS_URL=/sapmnt/software/sapinstall.params
 SAPINST_EXECUTE_PRODUCT_ID={{ProductId}} SAPINST_USE_HOSTNAME=${SAPAliasName}
 SAPINST_SKIP_DIALOGS="true" SAPINST_START_GUISERVER=false | tee -a $SSM_LOG_FILE
```

You can add additional sections in the SSM document to validate the SAP installation by checking the SAP process running on the host and sending the results to the SSM document log file. The following code is an example of how to do this:

```
- action: "aws:runShellScript"
  name: "Validate_Installation"
  inputs:
    runCommand:
    - #!/bin/bash
    - sid=`echo {{ Sid }} | tr '[:upper:]' '[:lower:]'}`
    - SID=`echo {{ Sid }} | tr '[:lower:]' '[:upper:]'}`
    - HOSTNAME=`hostname`
    - SIDADM=${sid}adm
    - su - $SIDADM -c "stopsap $HOSTNAME" | tee -a $SSM_LOG_FILE
    - su - $SIDADM -c "startsap $HOSTNAME" | tee -a $SSM_LOG_FILE
    - sleep 15
    - _SAP_UP=$(netstat -an | grep 3200 | grep tcp | grep LISTEN | wc -l )
    - echo "This is the value of SAP_UP - $_SAP_UP" | tee -a $SSM_LOG_FILE
    - if [ "$_SAP_UP" -eq 1 ]
    - then
    -   echo "$(date) __ done installing ASCS." | tee -a $SSM_LOG_FILE
    -   exit 0
    - else
    -   echo "$(date) __ ASCS could not be installed successfully. Fix the issue and
 rerun the automation" | tee -a $SSM_LOG_FILE
    -   exit 1
    - fi
- action: "aws:runShellScript"
  name: "Save_services_file"
  inputs:
    runCommand:
    - #!/bin/bash
    - grep -i sap /etc/services > /sapmnt/services
    - if [ -s /sapmnt/services ]
    - then
    -   echo "Services file copied to sapmnt" | tee -a $SSM_LOG_FILE
    -   exit 0
    - else
    -   echo "Services file could not be copied" | tee -a $SSM_LOG_FILE
    -   exit 1
    - fi
```

Tag the Systems Manager document

A tag is a label that you assign to an AWS resource. Each tag consists of a key and a value, both of which you define. For an overview of tagging Systems Manager resources, see Tagging Systems Manager resources in the *AWS Systems Manager User Guide*.

For detailed instructions on how to tag SSM documents, see Tagging Systems Manager documents in the *AWS Systems Manager User Guide*.

Example - tags and access management

You can use tagging for a variety of purposes. For example, if you're using AWS Identity and Access Management (IAM), you can control which users in your account can create, edit, or delete tags, and you can implement attribute-based access control (ABAC). For more information, see Grant permission to tag resources during creation and Control access to Amazon EC2 resources using resource tags in the *Amazon Elastic Compute Cloud User Guide for Linux Instances*.

Example - tags and billing

You can use tags to organize your AWS bill in a way that reflects your cost structure. To do this, sign up to get your AWS account bill with tag key values included. For more information about setting up a cost allocation report with tags, see Monthly cost allocation report in the *AWS Billing User Guide*. To see the cost of your combined resources, you can organize your billing information based on resources that have the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see Using cost allocation tags in the *AWS Billing User Guide*.

Automated operating system patching

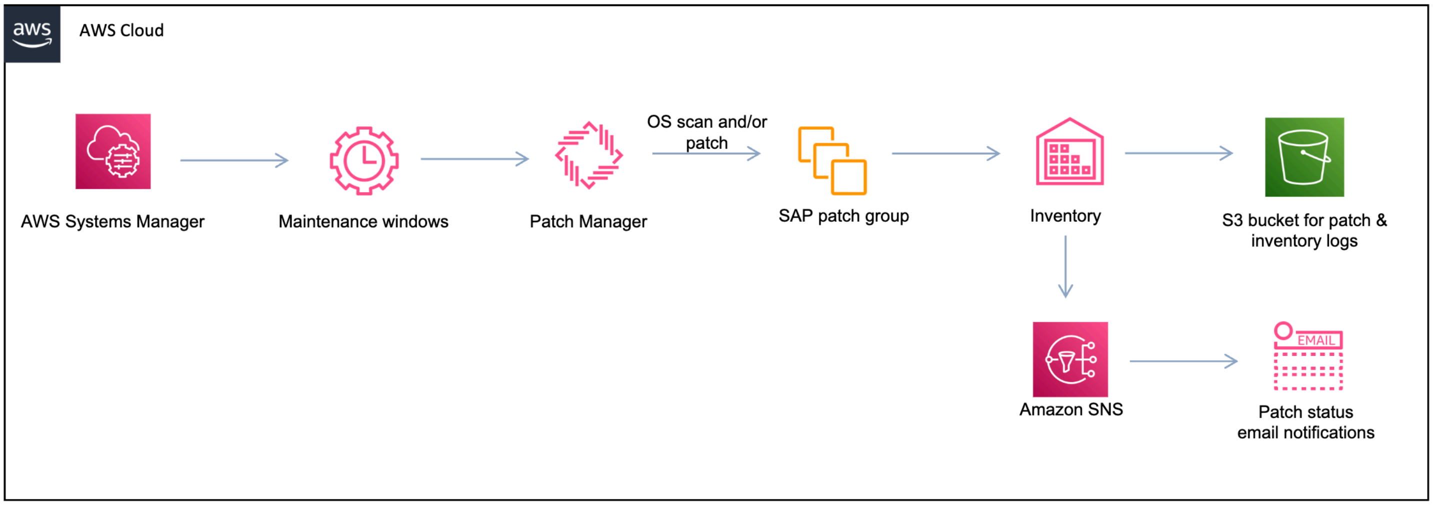
Patch management is an ongoing activity that is a key part of the SAP software lifecycle. It is a critical step in improving security, minimizing risk, remaining compliant, and reducing unplanned downtime. You can use AWS Systems Manager to automate system patching activities. Systems Manager can reduce the manual effort that is required to manage the SAP landscape, which saves time and IT resources.

An operating system patching strategy should adhere to SAP software lifecycle best practices. Patches should be applied downstream across the landscape, from development, to test, to production. This allows the patches to be tested in less-critical systems before deploying them into production. Because patching is a repeatable process, it can be automated with Systems Manager and can be documented as a standard operating procedures (SOP). This will ensure consistent patch management across the SAP landscape. The SOP should be updated continuously for future maintenance activities.

The sections below describe how to use Systems Manager to apply regular patches that are released by operating system vendors.

Automated operating system patching architecture

The diagram below highlights the AWS services that you can use to set up automated operating system patching and optional notifications on the patch status using Amazon Simple Notification Service (Amazon SNS).



The topics below contain descriptions of key components of the automated operating system patching setup. Familiarize yourself with them before continuing to the prerequisites.

Topics

- Patch Manager
- Lifecycle hooks

Patch Manager

Patch Manager is a capability of AWS Systems Manager that automates the process of patching managed nodes with security-related and general operating system updates. You can use Patch Manager to apply patches for operating systems and applications, such as installing service packs on Microsoft Windows nodes and performing minor version upgrades on Linux nodes.

Patch Manager helps to patch fleets of Amazon EC2 instances according to operating system type. This includes versions of Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), Oracle Linux, and Microsoft Windows Server that are supported by SAP on AWS. You can patch your instances on a schedule or on-demand by creating a patching configuration. You can also scan instances to see a report of missing patches or to automatically install missing patches.

Patch Manager integrates with AWS Identity and Access Management (IAM), Amazon CloudWatch Events, and AWS Security Hub to provide a secure patching experience that includes event notifications and the ability to audit usage.

Lifecycle hooks

Patch Manager allows you to add lifecycle hooks that enable a multi-step, custom patching process. These hooks let you perform a custom action on instances when the corresponding lifecycle event occurs.

When you patch the operating system of an SAP application, lifecycle hooks can help you perform SAP-specific operations and automate the operating system patching lifecycle. You can automate the following tasks using lifecycle hooks:

- Stop the SAP application and necessary database services
- Initiate database or storage snapshot backup
- Patch the operating system and reboot if necessary
- Start the SAP application and the database after successful operating system patch update

For more information about lifecycle hooks, see the following documentation:

- About the AWS-RunPatchBaselineWithHooks SSM document in the *AWS Systems Manager User Guide*
- Orchestrating multi-step in the *AWS Cloud Operations & Migrations Blog*

Automated operating system patching prerequisites

In addition to the prerequisites described in the Automation prerequisites section of this guide, verify the following prerequisites that are specific to automated operating system patching:

- Verify the Patch Manager prerequisites.

  Because the solution described here uses AWS Systems Manager Patch Manager, you must verify that you have satisfied all of the Patch Manager prerequisites. For more information, see Patch Manager prerequisites in the *AWS Systems Manager User Guide*.
- Ensure you have a backup of your SAP system.

  Before you make changes to the SAP system, verify that a backup is available to support rollback in case you encounter problems. You should have the following backups:

- Operating system backup – You should have an Amazon Machine Image (AMI) backup of the Amazon EC2 instance that consists of the base operating file system (`root` for Linux and `C:\` for Microsoft Windows) and the SAP application and database file systems.
- Database backup – If patching will occur on the database server, ensure you have the most recent database backup.

For data recovery recommendations, see Plan for data recovery in the *SAP Lens AWS Well-Architected Framework*.

Supported operating systems

The following operating systems are supported by SAP and Patch Manager. Check the Patch Manager prerequisites for currently supported versions of the operating systems. For more information, see Patch Manager prerequisites in the *AWS Systems Manager User Guide*.

- Oracle Linux

> **Note**
>
> Oracle Linux is required if you are running an Oracle database.

- Red Hat Enterprise Linux (RHEL)
- SUSE Linux Enterprise Server (SLES)
- Microsoft Windows Server

> **Note**
>
> - SUSE Linux and Red Hat Linux have SAP versions of the Linux operating system. SAP recommends that you use RHEL for SAP Solutions/Applications or SLES for SAP Applications to run the SAP application.
> - Oracle Linux operating system is required for Oracle Database Server and SAP NetWeaver Application Servers with Oracle client installed. For more information, see SAP Note 2358420 - Oracle Database Support for Amazon Web Services EC2 (SAP portal access required).

For each of these operating systems, you can bring your own subscription to AWS or use the Amazon Machine Images (AMIs) from the AWS Marketplace.

SAP Notes

Review the following SAP Notes. You require SAP portal access to check these references from SAP.

- SAP Note: 1656099 - SAP Applications on AWS: Supported DB/OS and Amazon EC2 products
- SAP Note: 2871484 - SAP supported variants of Red Hat Enterprise Linux
- SAP Note: 2358420 - Oracle Database Support for Amazon Web Services EC2
- SAP Note: 62988 - Service Packs for MS SQL Server
- SAP Note: 2235581 - SAP HANA: Supported Operating systems

Configuring automated operating system patching

The sections below contain detailed instructions on how to configure automated operating system patching.

Configure patch baselines

Patch Manager uses patch baselines, which include rules for auto-approving patches within days of their release, as well as a list of approved and rejected patches. For information about patch baselines, see About patch baselines in the *AWS Systems Manager User Guide*. You can use predefined patch baselines or create custom patch baselines. The sections below contain instructions on how to use both.

For information about patch baselines that is specific to Linux, see How patch baseline rules work on Linux-based systems in the *AWS Systems Manager User Guide*.

For information about the differences between Linux and Windows patching, see Key differences between Linux and Windows patching in the *AWS Systems Manager User Guide*. If your system landscape has a combination of Windows Server and Linux operating systems, such as Windows Server for SAP application servers and Linux for database servers, you can define a baseline for each operating system type.

Predefined patch baselines

Patch manager provides predefined patch baselines for each of the supported operating systems. If your patching requirement patches the predefined baseline configuration, you might be able to use a predefined patch baseline for operating system patching. Alternatively, you can create your own custom patch baselines. This gives you greater control over which patches are approved or rejected for your environment.

For information about predefined patch baselines, see Viewing AWS predefined patch baselines (console) in the *AWS Systems Manager User Guide*.

Note

SUSE Linux Enterprise Server for SAP Applications and Red Hat Enterprise Linux for SAP Applications require custom patch baselines.

The following table is a subset of the predefined patch baselines in the Patch Manager documentation. To view the full list of predefined patch baselines, see About predefined baselines in the *AWS Systems Manager User Guide*. The predefined patch baselines listed here are applicable to SAP.

Name	Supported operating system	Details
`AWS-OracleLinuxDefaultPatchBaseline`	Oracle Linux	Approves all operating system patches that are classified as "Security" and that have a severity level of "Important" or "Moderate". Also approves all patches that are classified as "Bugfix" 7 days after release. Patches are auto-approved 7 days after they are released or updated.[1]
`AWS-RedHatDefaultPatchBaseline`	Red Hat Enterprise Linux (RHEL)	Approves all operating system patches that are classified as "Security" and that have a severity level of "Critical" or "Important". Also approves all patches that are classified as "Bugfix". Patches are auto-

Name	Supported operating system	Details
		approved 7 days after they are released or updated.[1]
`AWS-SuseDefaultPatchBaseline`	SUSE Linux Enterprise Server (SLES)	Approves all operating system patches that are classified as "Security" and with a severity of "Critical" or "Important". Patches are auto-approved 7 days after they are released or updated.[1]
`AWS-DefaultPatchBaseline`	Windows Server	Approves all Windows Server operating system patches that are classified as "CriticalUpdates" or "Security Updates" and that have an MSRC severity of "Critical" or "Important". Patches are auto-approved 7 days after they are released or updated.[1]

[1] For Amazon Linux and Amazon Linux 2, the 7-day wait before patches are auto-approved is calculated from an `Updated Date` value in `updateinfo.xml`, not a `Release Date` value. Various factors can affect the `Updated Date` value. Other operating systems handle release and update dates differently. For information to help you avoid unexpected results with auto-approval delays, see How package release dates and update dates are calculated in the *AWS Systems Manager User Guide*.

Custom patch baselines

Unlike predefined patch baselines, custom patch baselines do not have default patch approvals and compliance levels. This gives you greater control over which patches are approved or rejected for your environment and allows you to define your custom repositories. For example, you can assign specific approval rules and compliance values. It is also possible to create a custom patch baseline by copying a predefined patch baseline and specifying the compliance values that you want to assign to patches.

You can use Patch Manager to create a custom patch baseline for Linux-based managed nodes, such as Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), Oracle Linux. You can also specify patch source repositories for each of these operating systems. See the sections below for additional information about patch sources for each.

For instructions on how to create a custom patch baseline for Linux and Windows, see the following documentation:

- Creating a custom patch baseline (Linux) in the *AWS Systems Manager User Guide*
- Creating a custom patch baseline (Windows) in the *AWS Systems Manager User Guide*

Patch sources

When you use the default repositories that are configured on a managed node for patching operations, Patch Manager scans for security-related patches or installs them. This is the default behavior for Patch Manager. On Linux systems, you can also use Patch Manager to install patches that aren’t related to security or that are in a different source repository than the default repository that is configured on the managed node.

In the procedure to create a custom patch baseline, there is an option to specify alternative patch source repositories if you are not using the default repository configuration. In each custom patch baseline, you can specify patch source configurations for up to 20 versions of a supported Linux operating system. For more information about alternative patch sources, see How to specify an alternative patch source repository (Linux) in the *AWS Systems Manager User Guide*.

> **Note**
>
> If you specify alternative repositories, you must also specify the default repositories as part of the alternative patch source configuration if you want those updates to be applied.

The sections below contain information about how to obtain patch source details for SLES for SAP Applications, RHEL for SAP Applications, and Oracle Linux. You can use this information to specify a patch source when you create a custom patch baseline.

Patch sources for SLES for SAP Applications

You can use one of the following patch repositories for SUSE Linux Enterprise Server (SLES) for SAP Applications:

- SUSE public cloud update infrastructure
- Private repository

  For information about how to use a private patch repository, see Private and local repositories in this guide.

The public cloud update infrastructure is a global network of update servers maintained by SUSE on AWS Cloud that provides low-latency access to patches from on-demand instances. Customers that use SUSE on-demand instances in AWS automatically connect to the public cloud update infrastructure on boot. You can view the SUSE patch source server details in the `/etc/hosts` directory.

You can connect to the public cloud update infrastructure through an internet gateway in a public subnet, NAT gateway in a private subnet, or through a local data center. To see the repository list, run the command `zypper ls`.

By default, all repositories are considered for patching. If you want to only patch certain repositories or if you are using multiple patch sources for repositories, you must explicitly add patch sources based on repository configuration.

Complete the following steps to identify the patch source for the repository that you would like to use for patching:

1. Navigate to the following directory to view the repository files:

   ```
   /etc/zypp/repos.d
   ```

2. Save the name and configuration for each repository file. For example, you might save the following:

   - Name – `SUSE_Linux_Enterprise_Server_for_SAP_Applications_x86_64:SLE-Product-SLES_SAPXX-SPX-Updates`
   - Configuration –

     ```
     name=SLE-Product-SLES_SAPXX-SPX-Updates
     enabled=1
     autorefresh=1
     baseurl=plugin:/susecloud?
     credentials=SUSE_Linux_Enterprise_Server_for_SAP_Applications_x86_64&path=/repo/
     SUSE/Updates/SLE-Product-SLES_SAP/XX-SPX/x86_64/update/
     ```

```
service=SUSE_Linux_Enterprise_Server_for_SAP_Applications_x86_64
```

3. Enter this information when you create the custom patch baseline in the **Patch sources** section of **Patch Manager**. For the full list of steps, see Creating a custom patch baseline (Linux) in the *AWS Systems Manager User Guide*.
4. If you add a patch source for any repository, you must add patch sources for all the repositories that you would like to patch, including the default repositories.

Important

Before you deploy the patch, you must accept the license agreement in the `zypper.conf` configuration file. You can find the file in the following directory:

```
/etc/zypp/zypper.conf
```

To accept the license agreement, uncomment the license agreement property and save it as:

```
autoAgreeWithLicenses = yes
```

Patch sources for RHEL for SAP Applications

You can use one of the following patch repositories for Red Hat Enterprise Linux (RHEL) for SAP Applications:

- Red Hat update infrastructure
- Local repository

  For information about how to use a private patch repository, see Private and local repositories in this guide.

Red Hat update infrastructure is a global network of update servers maintained by Red Hat on AWS Cloud that provides low-latency access to patches from on-demand instances. Customers that use Red Hat on-demand instances in AWS automatically connect to the Red Hat update infrastructure on boot.

The RHEL repositories are stored in the following location:

```
/etc/yum.repos.d/
```

Complete the following steps to identify the patch source for the repository that you would like to use for patching:

1. Run the following command to view the default, enabled repositories:

   ```
   cat /etc/yum.repos.d/* | grep -B 4 -A 6 "enabled=1"
   ```

   This command returns four lines before and six lines after each repository that is enabled. For example, the command might return something like this:

   ```
   [rhui-client-config-server-8-sap-bundle]
   name=Red Hat Update Infrastructure 3 Client Configuration for SAP Bundle
   mirrorlist=https://rhui3.REGION.ce.redhat.com/pulp/mirror/protected/rhui-client-
   config/rhel/server/8/$basearch/sap-bundle
   enabled=1
   gpgcheck=1
   gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
   sslverify=1
   sslcacert=/etc/pki/rhui/cdn.redhat.com-chain.crt
   sslclientcertexample=/etc/pki/rhui/product/rhui-client-config-server-8-sap-bundle.crt
   sslclientkeyexample=/etc/pki/rhui/rhui-client-config-server-8-sap-bundle.key
   ```

2. Save the name and configuration for each repository file. In this example, you would save the following:

   - Name – `rhui-client-config-server-8-sap-bundle`
   - Configuration

     ```
     name=Red Hat Update Infrastructure 3 Client Configuration for SAP Bundle
     mirrorlist=https://rhui3.REGION.ce.redhat.com/pulp/mirror/protected/rhui-client-
     config/rhel/server/8/$basearch/sap-bundle
     enabled=1
     gpgcheck=1
     gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
     sslverify=1
     sslcacertexample=/etc/pki/rhui/cdn.redhat.com-chain.crt
     ```

```
sslclientcertexample=/etc/pki/rhui/product/rhui-client-config-server-8-sap-
bundle.crt
```

3. For each entry that was returned by the command in the previous step, create a new patch source when you create a custom patch baseline in the **Patch sources** section of **Patch Manager**. For the full list of steps, see Creating a custom patch baseline (Linux) in the *AWS Systems Manager User Guide*.
4. If you add a patch source for any repository, you must add patch sources for all the repositories that you would like to patch, including the default repositories.

Patch sources for Oracle Linux

On Oracle Linux, the patch baseline uses preconfigured repositories on the managed node. All Oracle Linux Amazon Machine Images (AMIs) can access the public YUM repository. Only licensed Oracle Linux systems can access the Oracle ULN repository.

The Oracle Linux repositories are stored in the following location:

```
/etc/yum.repos.d/
```

Complete the following steps to identify the patch source for the repository that you would like to use for patching:

1. Run the following command to view the default, enabled repositories:

   ```
   cat /etc/yum.repos.d/* | grep -B 4 -A 6 "enabled=1"
   ```

   This command returns four lines before and six lines after each repository that is enabled. For example, the command might return something like this:

   ```
   [ol8-appstaream]
   name=Oracle Linux 8 Application Stream ($basearch)
   baseurl=https://yum$ociregion.$ocidomain/repo/OracleLinux/OL8/appstream/$basearch/
   gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
   gpgcheck=1
   ```

2. Save the name and configuration for each repository file. In this example, you would save the following:

   - Name – `ol8-appstaream`

- Configuration

```
name=Oracle Linux 8 Application Stream ($basearch)
baseurl=https://yum$ociregion.$ocidomain/repo/OracleLinux/OL8/appstream/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
```

3. For each entry that was returned by the command in the previous step, create a new patch source when you create a custom patch baseline in the **Patch sources** section of **Patch Manager**. For the full list of steps, see Creating a custom patch baseline (Linux) in the *AWS Systems Manager User Guide*.
4. If you add a patch source for any repository, you must add patch sources for all the repositories that you would like to patch, including the default repositories.

Oracle Linux 7 managed nodes use YUM as the package manager, while Oracle Linux 8 managed nodes use DNF as the package manager. Both package managers have an update notice, which is a file named `updateinfo.xml`. The update notice is a collection of packages that fix specific issues. Individual packages aren't assigned classifications or severity levels, so Patch Manager assigns the attributes of an update notice to the related packages and installs the packages based on the classification filters specified in the patch baseline.

Only patches specified in `updateinfo.xml` are applied if you are using the default patch baseline provided by AWS or if you do not select the option to include non-security update patches when you create a custom baseline. If you create a custom baseline and you do select the option to include non-security update patches, the patches in `updateinfo.xml` and the patches that are not in `updateinfo.xml` are applied. For more information, see How patch baseline rules work on Oracle Linux in the *AWS Systems Manager User Guide*.

Oracle Linux instances require internet access to the public YUM repository or Oracle ULN in order to download packages. If the Amazon EC2 instance is on a private subnet of an Amazon VPC, you can use a proxy server or a local YUM repository to download packages. For more information, see Configuring a System to Use a Proxy With a Yum Server in the Oracle documentation. Alternatively, Oracle Linux systems can work with Oracle Linux Manager for YUM package management. An Oracle Linux Manager system can be in a public subnet while Oracle Linux systems can be in a private subnet. For more information, see Oracle Linux Manager in the Oracle documentation.

Windows Server considerations

For additional information about security patches for Windows, see How security patches are selected and How patches are installed in the *AWS Systems Manager User Guide*.

Create patch groups

You can use patch groups to organize instances for patching. This can help you ensure that you only deploy patches to the correct set of instances and that the patches have been adequately tested before they are deployed. After you create the patch group, you can tag your Amazon EC2 instances to add them to the patch group and then add the patch group to a patch baseline.

You might want to organize patch groups by:

- Operating system – such as Linux and Windows
- Environment – such as development, test, and production
- Server function – such as SAP database servers and SAP application servers

> **Note**
>
> An Amazon EC2 instance can only be in one patch group at a time.

For more information about patch groups, see About patch groups in the *AWS Systems Manager User Guide*.

Tag Amazon EC2 instances to add to the patch group

After you create the patch group, use tags to add Amazon EC2 instances to the patch group. For detailed steps on how to do this, see Working with patch groups in the *AWS Systems Manager User Guide*.

Add the patch group to a patch baseline

To ensure that the correct patches are installed during the patching execution, you must register the patch group with a patch baseline. When the system applies a patch baseline to an instance, the service checks to see if a patch group is defined for the instance. For detailed steps on how to add a patch group to a patch baseline, see Add a patch group to a patch baseline in the *AWS Systems Manager User Guide*.

Note

Patch groups are not used in patching operations that are based on patch policies. For more information, see the following:

- Using Quick Setup patch policies
- Configure the home AWS Region
- Creating a patch policy

Applying patches

After you have created the patch baseline and tagged your Amazon EC2 instances to the patch group, you can apply patches. You can schedule patches or run them on-demand.

Scheduled patching

SAP maintenance activities are usually scheduled in advance. The non-critical SAP systems can be patched in an ad-hoc manner, such as a sandbox system. The patching process should be documented in runbooks. After the system is successfully patched, the patching activities for the downstream SAP systems can be scheduled, either using maintenance windows or directly from Patch Manager.

For more information about patching schedules, see the following documentation:

- About patching schedules using maintenance windows in the *AWS Systems Manager User Guide*
- Walkthrough: Creating a maintenance window for patching (console) in the *AWS Systems Manager User Guide*

On-demand patching

The **Patch now** option in Patch Manager allows you to run on-demand patching operations directly from the Systems Manager console. With this option, you do not need to create a schedule to update the compliance status of your managed nodes or to install patches on non-compliant nodes.

Scanning the Amazon EC2 instances allows you to identify systems that are potentially non-compliant, vulnerable, or un-patched. We recommend that you schedule system scans frequently, such as weekly.

For detailed instructions on how to run on-demand patching, see Patching managed nodes on demand in the *AWS Systems Manager User Guide*.

Patch summary

After the patch baseline has run, you can view the patch status in Patch Manager. For details about the patch summary and how to access it in Patch Manager, see Viewing patch Dashboard summaries (console) in the *AWS Systems Manager User Guide*.

Patch compliance reports

Patch compliance reports allow you to view the status of managed nodes. For more information about compliance reports, including detailed instructions on how to view them, see the following documentation:

- Working with patch compliance reports in the *AWS Systems Manager User Guide*
- Viewing patch compliance results (console) in the *AWS Systems Manager User Guide*

Monitoring

You can view Patch Manager output after each patch is run. By default, Patch Manager stores the first 48,000 characters of the command output. In some cases, you might want to view the complete log, such as for troubleshooting. In this case, the log output can be stored in Amazon S3. For details about how to store log output in Amazon S3, see Configuring Amazon CloudWatch Logs Logs for Run Command in the *AWS Systems Manager User Guide*.

Another option is to output the logs to Amazon CloudWatch Logs for unified logging. For more information, see Sending SSM Agent logs to CloudWatch Logs in the *AWS Systems Manager User Guide*.

For information about how to set up detailed monitoring and notifications, see Monitoring AWS Systems Manager in the *AWS Systems Manager User Guide*.

Private and local repositories

If you would like to manage your operating system repository locally, either within your VPC on AWS or an on-premises data center, without using an outbound internet connection for your instance, you can use a private or local repository.

Some reasons to use a private repository are:

- They provide access to repositories for Amazon EC2 instances that do not have access to the internet for security reasons.
- You have additional add-on products from vendors that are not provided through the public cloud update infrastructure.
- You want to deploy an organized and consistent set of patches across mission-critical workloads. Using an online repository might introduce new updates which could lead to inconsistency across the landscape.
- You want to improve software download times and reduce bandwidth overhead while patching a large fleet of infrastructure.

If you are on SUSE Linux Enterprise Server (SLES) and you want to use private repositories, make sure that the operating system repositories are pointing to the local repository instead of the respective vendor repositories before you use Patch Manager. If you are on Red Hat Enterprise Linux (RHEL) or Oracle Linux, you must use a custom baseline to point to local repositories.

Alternative tools for patching

In addition to AWS Systems Manager, there are other automated patching tools that you might use, which are listed below. This list is not exhaustive, but is meant to give you a starting point for doing your own research if you decide to consider alternate tools.

SUSE Manager

SUSE Manager is an infrastructure management tool for Linux systems. With SUSE manager, you can automate software management of SLES< RHEL and OEL operating systems. For more information, and a list of Amazon EC2 instances, see SUSE Manager 4.0 Documentation.

Repository Mirroring Tool (For SUSE Linux)

Repository Monitoring Tool (RMT) is a service from SUSE Linux that helps manage private repositories by downloading updates and distributing them across the landscape. This reduces network bandwidth usage and allows you to set more restrictive firewall policies. For more information, see the SUSE Linux Repository Mirroring Tool Guide.

Red Hat Satellite (For Red Hat Linux)

Red Hat Satellite is a system management solution that enables you to deploy, configure, and maintain your systems across physical, virtual, and cloud environments. Satellite Server synchronizes the content from the Red Hat Customer Portal and other sources, and provides functionality such as fine-grained lifecycle management, user and group role-based access control, integrated subscription management, as well as advanced GUI, CLI, or API access. For more information, see the Red Hat Customer Portal.

KernelCare (For Red Hat Linux)

KernelCare is a live patching system that patches Linux kernel vulnerabilities automatically, with no reboots. It works with all major Linux distributions, such as RHEL, CentOS, Amazon Linux, and Ubuntu. It also interoperates with common vulnerability scanners such as Nessus, Tenable, Rapid7, and Qualys. For more information, see KernelCare on AWS Marketplace.

Zypper Package Manager (For SUSE Linux)

Zypper is a command-line package manager for installing updating, and removing packages. It can also be used to manage repositories. Zypper offers advantages over graphical package managers such as scripting actions. For more information, see the Zypper package manager documentation.

Considerations for multiple accounts

When you run SAP workloads in AWS, you must consider an AWS account strategy that meets the security controls of your organization. For example, you might separate SAP from non-SAP workloads and separate production from non-production environments. AWS Systems Manager does not support multi-account patching.

In every AWS account with SAP workloads, patch baselines should be created and patch execution should be performed to ensure that patching is applied to all SAP systems. In a multi-account environment, this should also follow the SAP best practice of patching in the development account, then test, and finally in the production AWS account.

Automation troubleshooting

If you encounter errors related to SAP automation, refer to the Troubleshooting Systems Manager Automation documentation in the *AWS Systems Manager User Guide*. There, you will find an action-specific failures reference as well as information about common errors such as access denied errors and errors related to timed out or failed statuses after the execution started.

Logging installation steps

You can log individual automated installation steps with the code below. In this example, logs are added to `$SSM_LOG_FILE` for each `run` command.

```
action: "aws:runShellScript"
name: "Validate_Installation"
inputs:
runCommand:
- #!/bin/bash
- sid=echo {{ Sid }} | tr '[:upper:]' '[:lower:]'}``
- SID=echo {{ Sid }} | tr '[:lower:]' '[:upper:]'}``
- HOSTNAME=hostname``
- SIDADM=${sid}adm
- su - $SIDADM -c "stopsap $HOSTNAME" | tee -a $SSM_LOG_FILE
- su - $SIDADM -c "startsap $HOSTNAME" | tee -a $SSM_LOG_FILE
- sleep 15
- _SAP_UP=$(netstat -an | grep 3200 | grep tcp | grep LISTEN | wc -l )
- echo "This is the value of SAP_UP - $_SAP_UP" | tee -a $SSM_LOG_FILE
- if [ "$_SAP_UP" -eq 1 ]
- then
-   echo "$(date) __ done installing ASCS." | tee -a $SSM_LOG_FILE
-   exit 0
- else
-   echo "$(date) __ ASCS could not be installed successfully. Fix the issue and rerun
 the automation" | tee -a $SSM_LOG_FILE
-   exit 1
- fi
```