



SAP BusinessObjects Guides

SAP BusinessObjects on AWS



SAP BusinessObjects on AWS: SAP BusinessObjects Guides

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Home	1
SAP BOBI Platform on AWS Deployment and Operations Guide for Linux	2
Overview	2
Prerequisites	2
General AWS Knowledge	3
Recommended Reading	3
Technical Requirements	4
Planning	6
Choosing a Region	6
Choosing an Availability Zone	7
Architecture Options	7
Sizing	10
High Availability (HA) and Disaster Recovery (DR)	10
Security & Compliance	10
Operating System	13
Compute	14
Network	14
Storage	14
Deployment	16
Step 1. Prepare Your AWS Account	17
Step 2. Create a JSON file for the Amazon EBS storage	17
Step 3. Launch the Amazon EC2 Instance	18
Step 4. Prepare the EC2 Instances	19
Step 5. Create Amazon EFS Mount for FileStore	23
Step 6. Prepare and Install the CMS Database (Only for RDS Database)	26
Step 7. Install CMS Database (Only for CMS Database on EC2 Instance)	27
Step 8. Install SAP BOBI Platform Nodes	27
Step 9. Configure End User Access for Multi-Node Deployment	28
Operations	29
Tagging AWS Resources	29
Monitoring	30
Operating System Maintenance	30
Backup and Restore	31
Compute	34

Cost Optimization	35
Automation	35
Integration with AWS Big Data Services	35
Support	36
Document Revisions	36
SAP BOBI Platform on AWS Deployment and Operations Guide for Windows	37
Overview	37
General AWS Knowledge	38
Prerequisites	38
Recommended Reading	38
Technical Requirements	39
Planning	41
Choosing a Region	41
Choosing an Availability Zone	42
Architecture Options	42
Storage	45
Sizing	45
High Availability (HA) and Disaster Recovery (DR)	46
Security & Compliance	46
Operating System	48
Compute	49
Network	49
Storage Services	49
Deployment	51
Step 1. Prepare Your AWS Account	51
Step 2. Create a JSON file for the Amazon EBS storage	52
Step 3. Launch the Amazon EC2 Instance	52
Step 4. Prepare Each EC2 Instance for SAP Installation	53
Step 5. Create Amazon FSx Volumes	54
Step 6. Prepare and Install the CMS Database (Only for RDS Database)	54
Step 7. Install CMS Database (Only for CMS Database on EC2 Instance)	55
Step 8. Install SAP BOBI Platform Nodes	55
Step 9. Configure End User Access for Multi-Node Deployment	56
Operations	57
Tagging AWS Resources	57
Monitoring	57

Operating System Maintenance	57
Backup and Restore	59
Compute	62
Cost Optimization	63
Automation	63
Integration with AWS Big Data Services	64
Support	64
Document Revisions	64
SAP BusinessObjects BI Platform HA/DR Guide for Linux	65
.....	66
Prerequisite knowledge	66
AWS services	66
SAP BusinessObjects BI Platform on AWS	66
High availability	67
HA for SAP BusinessObjects BI Platform on AWS	68
Planning the deployment in a primary region	69
Installing SAP BusinessObjects BI Platform for HA	77
HA for SAP Data Services	83
HA testing	83
Disaster recovery	84
DR for SAP BusinessObjects BI Platform on AWS	84
Planning the deployment in the DR region	84
Installing SAP BusinessObjects BI Platform in the DR region	90
DR for SAP Data Services	92
Summary	92
Additional tips	93
Tagging AWS resources	93
Third-party software components	93
SAP BusinessObjects BI Platform HA/DR Guide for Windows	94
.....	95
Prerequisite knowledge	95
AWS services	95
SAP BusinessObjects BI Platform on AWS	95
High availability	96
HA for SAP BusinessObjects BI Platform on AWS	97
Planning the deployment in a primary region	98

Installing SAP BusinessObjects BI Platform for HA	106
HA for SAP Data Services	111
HA testing	112
Disaster recovery	112
DR for SAP BusinessObjects BI Platform on AWS	113
Planning the deployment in the DR region	113
Installing SAP BusinessObjects BI Platform in the DR region	119
DR for SAP Data Services	121
Summary	121
Additional tips	122
Tagging AWS resources	122
Third-party software components	122

SAP BusinessObjects Guides

This section covers the following guides.

- [Deployment and operations on Linux](#)
- [Deployment and operations on Windows](#)
- [High availability and disaster recovery on Linux](#)
- [High availability and disaster recovery on Windows](#)

Additional SAP on AWS documentation

- [General SAP guides](#)
- [SAP HANA on AWS](#)
- [SAP NetWeaver on AWS](#)
- [Databases for SAP applications on AWS](#)
- [AWS Launch Wizard for SAP](#)
- [AWS Systems Manager for SAP](#)
- [AWS SDK for SAP ABAP](#)
- [AWS Migration Hub Orchestrator](#)

SAP BusinessObjects Business Intelligence Platform on AWS Deployment and Operations Guide for Linux

SAP specialists, Amazon Web Services

Last updated: January 2023

The purpose of this guide is to provide an overview of how to implement and operate SAP BusinessObjects (BO) Business Intelligence (BI) Platform (also referred in this document as SAP BOBI Platform) on Amazon Elastic Compute Cloud (Amazon EC2).

This guide is not intended to replace the standard SAP BOBI Platform installation and administration guides, operating system, or relational database management system (RDBMS) documentation.

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](#).

The purpose of this guide is to provide an overview of how to implement and operate SAP BusinessObjects (BO) Business Intelligence (BI) Platform (also referred in this document as SAP BOBI Platform) on Amazon Elastic Compute Cloud (Amazon EC2). This guide covers common AWS services and features that are relevant for SAP BusinessObjects BI platform. This guide is not an exhaustive list of all possible configuration options. It covers solutions common to typical deployment scenarios.

This guide is not intended to replace the standard SAP BOBI Platform installation and administration guides, operating system, or relational database management system (RDBMS) documentation.

Prerequisites

Before you start implementing your SAP BOBI Platform systems, we recommend that you review these prerequisites to ensure there are minimal interruptions and delays.

General AWS 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, see [Getting Started with AWS](#).)

- [Amazon ECS](#)
- [Amazon VPC](#)
- [AWS Identity and Access Management](#)
- [Amazon EBS](#)
- [Amazon S3](#)
- [AWS Systems Manager](#)
- [AWS CloudFormation](#)
- [Amazon FSx for NetApp ONTAP](#)

Recommended Reading

We also recommend you first read some key overview and best practice guides:

- [SAP on AWS Overview and Planning Guide](#)
- [Getting Started with Architecting SAP on the AWS Cloud](#)
- [Best Practices for Linux on Amazon EC2](#)

SAP Notes

The SAP notes listed in Table 1 have useful information regarding SAP BOBI deployment in AWS.

Table 1: SAP Notes for SAP BOBI deployment on AWS

SAP Note	Description
1588667	SAP on AWS: Overview of related SAP notes and web links
1656099	SAP on AWS: Supported products, platforms, and landscapes

SAP Note	Description
2442979	Amazon S3 recommendations for SAP BusinessObjects Business Intelligence Platform
2438592	BI Platform 4.2 Cloud Support

Technical Requirements

- Ensure that any services you will use for your SAP BOBI Platform deployment are not constrained by default AWS service limits. You can find the details at [Service endpoints and quotas](#). You can increase soft limits by submitting a support ticket to AWS.
- Make sure that the following information is available in relevance to your existing AWS resources. You will need this information while executing AWS Command Line Interface (AWS CLI) commands to create your Amazon EC2 and Amazon Elastic Block Store (Amazon EBS) resources:

Information You Need	Description
Region ID	AWS Region where you want to deploy your AWS resources.
Availability Zone	Availability Zone within your target region where you want to deploy your resources.
Amazon VPC ID	Amazon Virtual Private Cloud (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 latest Linux AMIs on AWS Marketplace .

Information You Need	Description
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.

- Ensure that you have a key pair that you can use to launch your Amazon EC2 instances. See [Amazon EC2 Key Pairs for Linux Instances](#) if you need to create a key.
- Ensure that you have the network details like VPC ID, Subnet ID, and so on, of the VPC where you plan to launch your Amazon EC2 instances to host your SAP BOBI Platform applications.
- Ensure that the required ports are open on the security group attached to your Amazon EC2 instance to allow log in to the operating system.
- For distributed or high availability (HA) installations, ensure that the security group attached to each application servers allows communication over the required ports between them. The easiest way to do this is to create a rule that references a security group as its own source and allow traffic on the required ports for that rule.
- If you intend to use the AWS CLI to launch your instances, then ensure that you have installed and configured AWS CLI with the appropriate credentials. See [Configuring the AWS CLI](#) for more details.
- If you intend to use the AWS Management Console to launch your instances, then ensure that your IAM user has permission to launch and configure Amazon EC2, Amazon EBS, and so on. See the [IAM documentation](#) for more details.
- Ensure that you have the required SAP software available either via an Amazon Simple Storage Service (Amazon S3) bucket or on a file share accessible from an Amazon EC2 instance. If you use Amazon S3, make sure to assign appropriate IAM role permissions to the EC2 instance to allow S3 access.

- All enterprise customers use DNS service. You can create a hosted zone in Amazon Route 53. You can optionally use AWS Directory Service for Microsoft Active Directory, also known as AWS Managed Microsoft AD. This service lets your directory-aware workloads and AWS resources use managed Active Directory in the AWS Cloud. For more details on this service, see [AWS Directory Service](#) and [Create Your AWS Managed Microsoft AD directory](#).

Planning

The following topics are important for planning the SAP BOBI Platform in AWS Cloud.

Topics

- [Choosing a Region](#)
- [Choosing an Availability Zone](#)
- [Architecture Options](#)
- [Sizing](#)
- [High Availability \(HA\) and Disaster Recovery \(DR\)](#)
- [Security & Compliance](#)
- [Operating System](#)
- [Compute](#)
- [Network](#)
- [Storage](#)

Choosing a Region

When choosing which AWS Region to deploy your SAP environment in you should consider the following topics:

- Proximity to your on-premises data centers, systems, and end users to minimize network latency.
- Data residency and compliance requirements.
- Whether the AWS products and services you plan to use are available in the Region. For a detailed list of AWS products and services by Region, see the [Region Table](#) page on the AWS website.
- The Amazon EC2 instances types you plan to use are available in the region. To view AWS Region availability for a specific instance type, see the [Amazon EC2 Instance Types for SAP](#) page.

Choosing an Availability Zone

No special considerations are required when choosing an Availability Zone for your SAP deployment on AWS. If high availability (HA) is a requirement, use multiple Availability Zones. For more information about HA, see the [SAP BusinessObjects BI Platform on AWS: HA/DR Guide for Linux](#).

For more information about AWS Regions and Availability Zones, see [AWS Global Infrastructure](#).

Architecture Options

The server-side architecture of SAP BOBI Platform consists of five tiers: web, management, storage, processing, and data. (For details, see the administrator's guide on the [SAP BusinessObjects Business Intelligence Platform](#) website). The following list provides high-level details.

- **Management tier:** Includes the CMS servers, event servers, and associated services.
- **Storage tier:** Includes input and output file repository servers. The file system used by these servers to store files, such as documents, reports, and universes, must be on a shared file system.
- **Web tier and processing tier:** Performs functions like receiving and processing user requests.
- **Data tier:** Consists of the CMS system database and the auditing data store.

You can have following example architecture designs for the above tiers:

- Install all tiers on the same EC2 instance.
- Install the application and database tiers on two separate EC2 instances.
- Install different tiers on multiple EC2 instances grouped based on customer-specific requirements.

The architecture choice depends on multiple factors like complexity, cost, sizing, and technical restrictions. For example, if you use [Amazon RDS](#) as the database, application tiers cannot be installed with database.

CMS and Audit Database Architecture Options

You have the choice of deploying the SAP BOBI Platform application on a standard SAP supported database like SAP HANA, SAP ASE, IBM DB2, Microsoft SQL Server, or [Amazon Relational Database](#)

[Service \(Amazon RDS\)](#). For supported [Amazon RDS](#) database types, see [SAP Note 1656099 SAP on AWS: Supported SAP, DB/OS and AWS EC2 products](#).

[Amazon RDS](#) is a service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. Amazon RDS takes over many of the difficult or tedious management tasks such as backups, software patching, automatic failure detection, and recovery. You can read more about this service in [Amazon RDS documentation](#).

Figure 1 shows an example large scale architecture of SAP BOBI with multi-AZ and multi-instance architecture. Web, Management, Processing, and Data tiers are all distributed on different EC2 instances. [Amazon RDS](#) MySQL is used for CMS database.

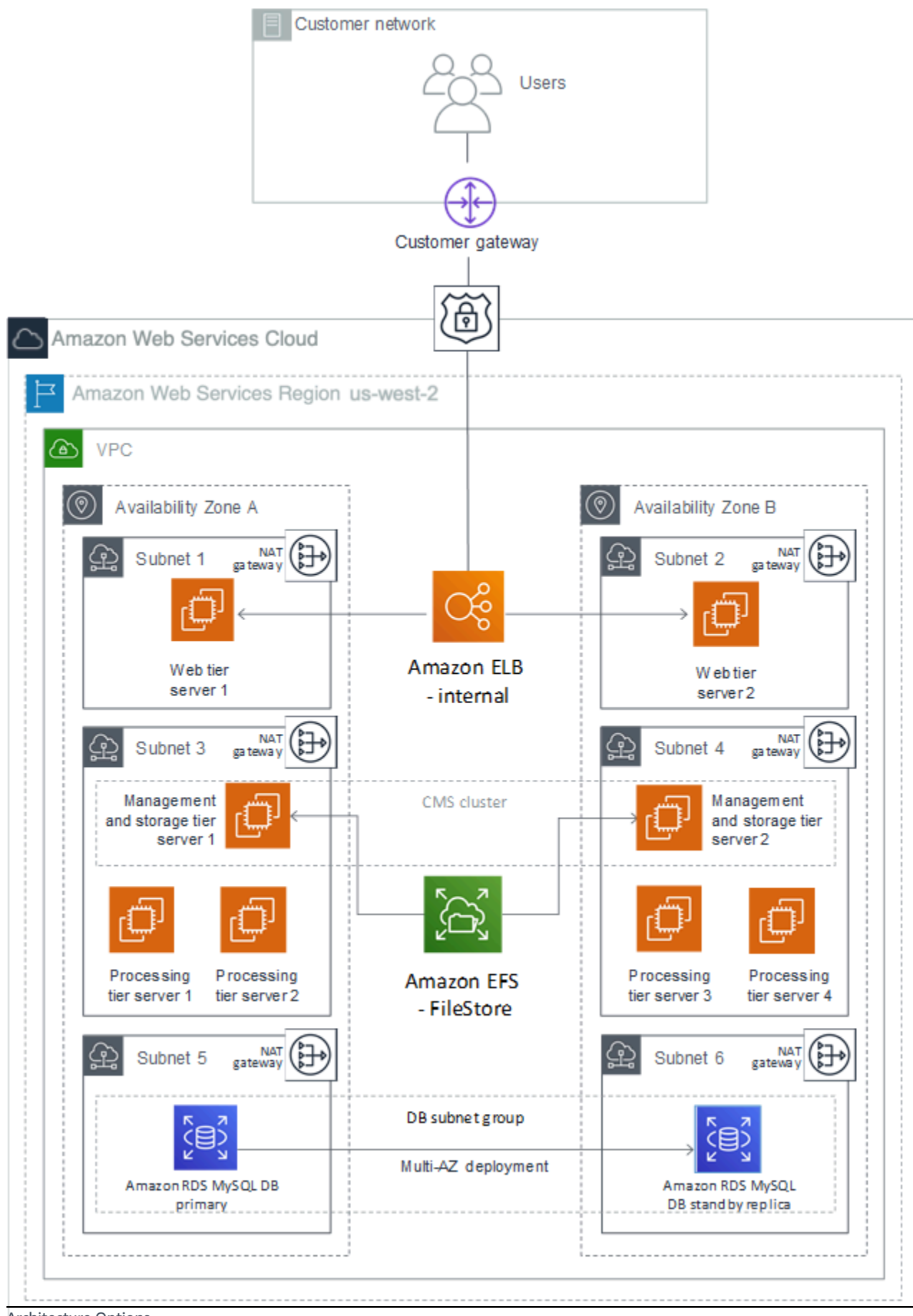


Figure 1: SAP BOBI with multi-AZ and multi-instance architecture

Sizing

At a high level, BOBI platform sizing is a two-step process. The first step is to get SAPS through the SAP Sizing tool [Quick Sizer](#). The second step is to map the output to appropriate Amazon EC2 instance types. See the SAP BOBI sizing guide available from the [SAP BusinessObjects Business Intelligence Platform help documentation](#). Follow the standard SAP Quick Sizer to determine the right SAPS requirements for your workload. Once you have identified the SAPS numbers, you can use any of the AWS [SAP certified instances](#) for your SAP BOBI Platform instances. Make sure to read the SAP BusinessObjects Business Intelligence section of the note carefully to ensure that you meet the version, EC2 resource, operating system, and database requirements for AWS.

For sizing your storage requirements for the database and app tier, AWS provides various volume types, such as general purpose SSDs (gp3) and provisioned IOPS SSD (io2). You can read more about the differences at [Amazon EBS Volume Types](#). As general guidance, we recommend that you consider the gp3 volume type as a starting point to see if it satisfies your throughput and IOPS requirement of your workload, and consider io2 volume types if you have a need for sustained IOPS.

It is easy to add capacity to your existing SAP BOBI deployment in AWS. Therefore, for migrating your existing deployment to AWS, you may not need the same degree of over-provisioning as is typical for on-premises environments.

High Availability (HA) and Disaster Recovery (DR)

If you require a highly available BOBI environment, then it critical to design the HA and DR environment that can support the recovery time objective (RTO) and recovery point objective (RPO) that your business teams have established. For more information, see the [SAP BusinessObjects BI Platform on AWS: HA/DR Guide for Linux](#).

Security & Compliance

The following AWS security resources help you achieve the level of security you require for your SAP NetWeaver environment on AWS:

- [AWS Cloud Security](#)
- [CIS AWS Foundations Benchmark](#)

- [Introduction to AWS Security](#)
- [Best Practices for Security, Identity, & Compliance](#)
- [AWS Well-Architected Framework Security Pillar](#)
- [Network and security features for Linux and Amazon EC2](#)

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 Linux EC2 instances:

- 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 Amazon EC2 instances and the security state of your applications running on the instances.

Encryption

Security is a priority on AWS. 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 in transit 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 they require no additional action from you or your applications.

Amazon EFS supports two forms of encryption for file systems: encryption of data in transit and encryption of data at rest. You can enable encryption of data at rest when creating an Amazon EFS file system. You can enable encryption of data in transit when you mount the file system to protect data at rest by using either server-side encryption or client-side encryption.

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

- [Encrypting Amazon EFS Data at Rest and Data in Transit](#)
- [Protecting Amazon S3 Data Using Encryption](#)
- [Amazon EBS Encryption](#)

Security Groups/Network ACLs

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 you scale workloads horizontally or require high availability, you may choose to include 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 you use Linux, some configuration changes may be necessary in the security groups, route tables, and network ACLs. You can refer to the operating system product documentation, or other sources such as the [Security Group Rules Reference](#) in the Amazon Elastic Compute Cloud (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 may set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

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 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 AWS 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 produced by CloudTrail enables 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 set up notifications on SSH login to your email address or mobile phone.

Operating System

Customers can choose to bring their license subscriptions or use AWS Marketplace to purchase licenses. Operating systems such as [SUSE Linux Enterprise Server for SAP](#) and [Red Hat Enterprise Linux for SAP with HA and Update Services](#) are optimized for running SAP and come with high availability solution for SAP NetWeaver and SAP HANA database. For supported operating systems, see [SAP Note 1656099](#).

SLES

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. For details, see [SUSE Linux Enterprise Server on Amazon EC2 - FAQs](#).

RHEL

If you plan to use the BYOS model with RHEL and the Red Hat Cloud Access Gold Images, ensure your subscription has access to the [Red Hat Cloud Access](#) program. For details, see [Red Hat Enterprise Linux on Amazon EC2 - FAQs](#) and [Red Hat Cloud Access](#).

Amazon Machine Image (AMI)

A base AMI is required to launch an Amazon EC2 instance. Depending on your choice of operating system, ensure that you have access to the appropriate AMI in your target Region for the deployment.

If you are using AWS CLI, you must provide the AMI ID when you launch the instance.

Compute

AWS has certified multiple instance families with different sizes to run SAP workloads. The details of the latest list of EC2 instance types certified are all specified in our webpage [Amazon EC2 Instance Types for SAP](#).

Select the appropriate EC2 instance type based on your requirements as per the sizing section. AWS recommends you use the latest generation of your selected instance family where possible (assuming it is SAP certified and therefore listed in the above webpage).

Network

Ensure that you have your network constructs set up to deploy resources related to your SAP workload. If you haven't already set up network components like Amazon Virtual Private Cloud (Amazon VPC), subnets, route tables and so on., you can use the [AWS Quick Start for Modular and Scalable VPC Architecture](#) to easily deploy scalable VPC architecture in minutes. Refer to the deployment guide for more details, then set up your EC2 instances for the SAP workload within this VPC.

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

Storage

The SAP BOBI Platform uses the following AWS storage services:

- [Amazon Elastic Block Store \(Amazon EBS\)](#) is used for block storage requirements of SAP BOBI Platform application servers and databases (when the database is installed on EC2).

Figure 2 shows an example use of EBS volumes for application and database. In this example, EBS volumes are used for root volumes, SAP BOBI Platform installation directory, operating system swap volume, and database data and log volumes. The CMS database is typically a small database that stores information like users, SAP BOBI Platform servers, folders, and other configurations. Therefore, it does not have the same storage performance requirements as other enterprise OLTP/OLAP databases. Follow the best practices of the database vendor for designing storage for the SAP BOBI Platform database.

- [Amazon Elastic File System \(Amazon EFS\)](#) is used for shared file system requirements of SAP BOBI Platform application servers installed on Linux EC2 instances. The FileStore in an SAP BOBI

Environment requires a shared file system as it stores the content like reports, universes, and connections, which are used by all application servers of that system.

- [Amazon Simple Storage Service \(Amazon S3\)](#) is used for storing the backups of SAP BOBI Platform application servers.

Figure 2 shows an example use of AWS storage services by an SAP BOBI Platform installation. In this example, two SAP applications servers and a database are installed on three separate EC2 instances with Linux operating system. EBS volumes are used for local file systems like root, install, swap, data, and log volumes. Amazon EFS is used for shared file system FileStore.

 **Note**

For SAP BusinessObjects file storage, you can use Amazon FSx for NetApp ONTAP to store your content in a shared file system. You can also use FSx for ONTAP file system for your CMS database data and log volumes. For more information, see [Amazon FSx for NetApp ONTAP](#).

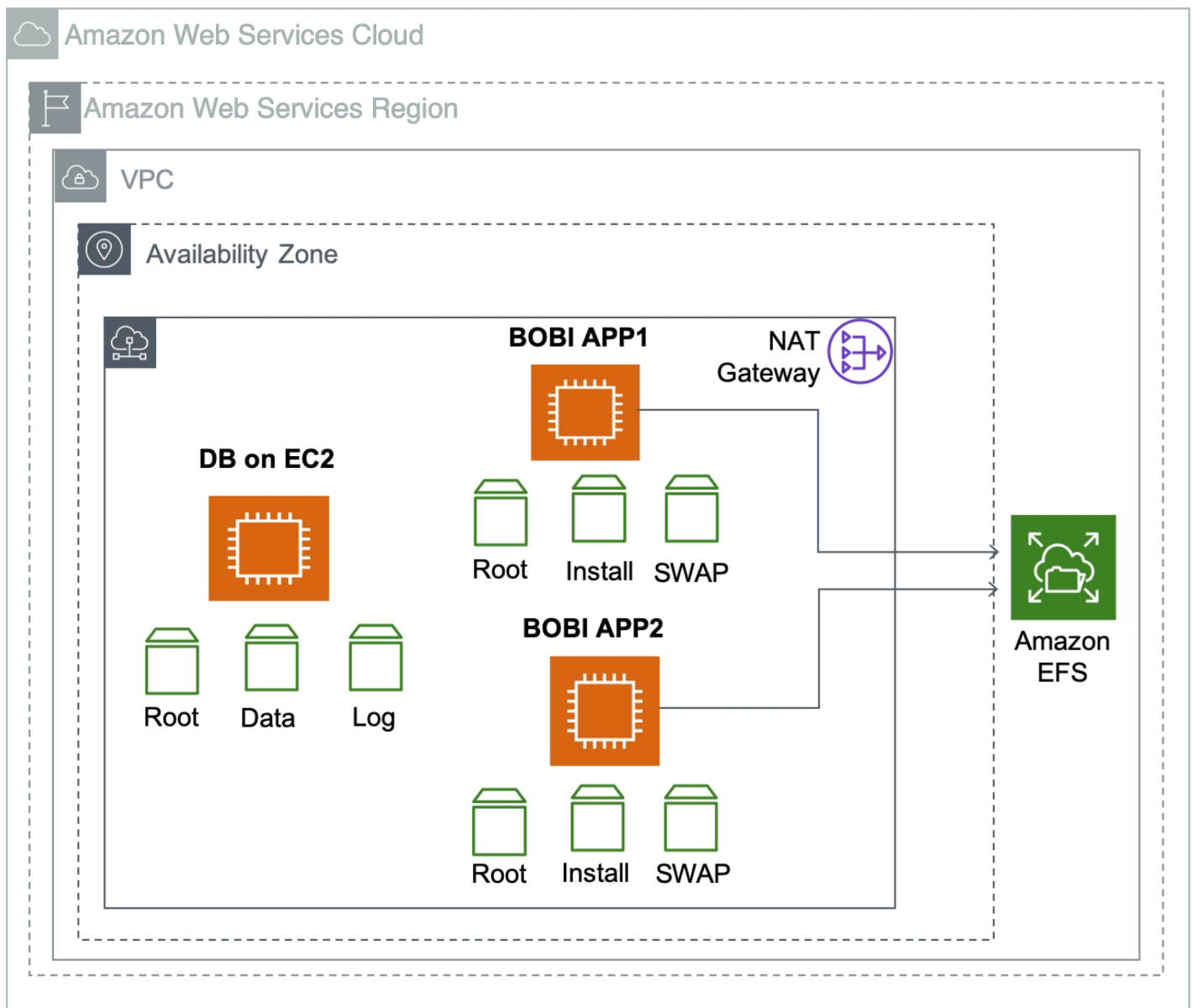


Figure 2: AWS storage system use on SAP BOBI Platform installation

Deployment

In this deployment, we will provision an Amazon EC2 instance for installing the SAP application servers and the CMS database (if you are using database on EC2). When using Amazon RDS for CMS database, follow [Step 7. \(Only for CMS Database on EC2 Instance\) Installing CMS Database](#).

In this deployment, we will provision an Amazon EC2 instance for installing a standalone Oracle database standard system.

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 see the documentation.

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 our scenario, we are using [AWS Launch Wizard for SAP](#) in a single-AZ deployment to create the VPC, subnets, security groups, and IAM roles. This is just an example setup and customers should follow their own network layout and comply with their own security standards. This may include:

- using AWS Launch Wizard for SAP in for multi-AZ deployment of SAP HANA
- using a landing zone solution like [AWS Control Tower](#)
- work with their cloud team (for example a Cloud Center of Excellence or CCoE) to use existing standards

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

Note

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

2. 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 the SAP workload.

Step 2. Create a JSON file for the Amazon EBS storage

Create a JSON file that contains the storage requirements for SAP BOBI Platform server volumes.

Below is an example JSON file with two EBS volumes for swap and SAP BOBI Platform installation directory. You can modify this file as per your requirements:

```
[
  {
    "DeviceName": "/dev/sdh",
    "Ebs": {
      "VolumeSize": 32,
      "VolumeType": "gp32",
      "DeleteOnTermination": true
    }
  },
  {
    "DeviceName": "/dev/sdg",
    "Ebs": {
      "VolumeSize": 50,
      "VolumeType": "gp32",
      "DeleteOnTermination": true
    }
  }
]
```

Step 3. Launch the Amazon EC2 Instance

Launch the Amazon EC2 instance for the SAP BOBI Platform 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.

```
$ 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. Create a JSON file for the Amazon EBS storage](#).

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-xxxxxxxxxxxxxxxx --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 EC2 instances using the AWS Management Console. For detailed steps, see [Launch Linux EC2 Instances using AWS Management Console](#).

Step 4. Prepare the EC2 Instances

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 DNS name according to your requirements. For detailed steps, see the AWS Knowledge Center article for your operating system:

- [Assign a static hostname to Amazon EC2 instance running SuSe Linux](#)
- [Assign a static hostname to Amazon EC2 instance running on RHEL](#)

Alternatively, you can edit the `/etc/hosts` file and manually add this entry. For SAP systems, the maximum length of the hostname should not exceed 13 characters. The name should comply with SAP standards. See [SAP OSS Note 611361](#) for details (requires access to SAP Service Marketplace).

Install Prerequisite Packages

Note

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

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

- SUSE syntax:

To install a package: `zypper -n 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`

2. Install `nfs-utils`, which is required for mounting the Amazon EFS mounts onto the Linux host.

- SUSE command:

```
# zypper install nfs-utils
```

- RHEL command:

```
# yum install nfs-utils
```

3. Install `nvme-cli` package to view the NVME device mapping of Amazon EBS volumes
4. Install SSM Agent by following the instructions in the [Systems Manager user guide](#).
5. Install the [AWS Data Provider for SAP](#).

```
#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
```

Identify Amazon EBS Device from NVMe Block Devices

On [Nitro-based](#) instances, the device name 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.

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 it by label than the actual NVME ids. This will also help in situation where you need to change your instance type between Nitro-based and non Nitro-based instances.

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
nobarrier,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
#
```

In the example code above, you can see that `/usr/sap` is mounted on device `/dev/nvme1n1`.

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 5. Create Amazon EFS Mount for FileStore

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 this 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 output:

```
{
  "SizeInBytes": {
    "Value": 0
  },
  "CreationToken": "efsforsap",
  "Encrypted": false,
```

```
"CreationTime": 1523374253.0,  
"PerformanceMode": "generalPurpose",  
"FileSystemId": "fs-abc12def",  
"NumberOfMountTargets": 0,  
"LifeCycleState": "creating",  
"OwnerId": "xxxxxxxxxxxxx"  
}
```

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

4. Create the tag for the `FileSystemId`:

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

For example:

```
$ aws efs create-tags --file-system-id fs-abc12def --tags Key=filestore,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": "xxxxxxxxxxxx",  
  "FileSystemId": "fs-abc12def",  
  "LifeCycleState": "creating",  
  "SubnetId": "xxxxxxxxxxxx",  
  "OwnerId": "xxxxxxxxxxxx",  
  "IpAddress": "x.x.x.x"  
}
```

Make a note of the `LifeCycleState`, 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": "xxxxxxxxxx ",
      "OwnerId": "xxxxxxxxxx",
      "IpAddress": "x.x.x.x"
    }
  ]
}
```

7. Mount EFS file system using DNS name or its IP address and create folders for mounting EFS.

Tip

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

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

For example:

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

Alternatively, you can also use the `IpAddress` from step 11. In this example, we use the `IpAddress` instead.

```
# mkdir /test
# sudo mount -t nfs -o
  nfsvers=4.1,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2 <ip-address>:/ /
test
# cd /test
# mkdir FileStore
# ls -ltr
```

```
drwxr-xr-x 2 root root 6144 Apr  1 16:08 FileStore
# cd /
# umount /test
```

8. Create the mount points for FileStore (used here as an example for the name of your FileStore mount point):

```
# mkdir /prdbobi_fs
```

9. Mount the Amazon EFS file system using DNS name of EFS.

```
# sudo mount -t nfs -o
nfsvers=4.1,rsiz=1048576,wsiz=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,rsiz=1048576,wsiz=1048576,hard,timeo=600,retrans=2 x.x.x.x:/TRANS /
usr/sap/trans
# sudo mount -t nfs -o
nfsvers=4.1,rsiz=1048576,wsiz=1048576,hard,timeo=600,retrans=2 x.x.x.x:/TRANS /
usr/sap/trans
```

Step 6. Prepare and Install the CMS Database (Only for RDS Database)

This option is applicable only when Amazon RDS MySQL is used for the CMS database. You can create a separate database for the auditing database if it's required.

1. Create a DB subnet group for an RDS instance by following the instructions in [Create a DB Subnet Group](#).
2. In the [Amazon RDS console](#), launch an Amazon RDS MySQL DB instance by following the instructions in the [Creating a DB Instance Running the MySQL Database Engine](#).
3. Choose a supported DB version based on [SAP Note 1656099 - SAP on AWS: Supported SAP, DB/OS and Amazon EC2 products](#), and select the instance type and storage based on your sizing output.
4. On the **Specify DB details** page, in the **Instance specifications** section, choose **Create replica in different zone**.

5. The **Choose use case** page asks if you are planning to use the DB instance you are creating for production. If you choose **Production - MySQL**, the Multi-AZ failover option is preselected. You can deselect this option if you are not installing a highly available system.
6. On the **Configure advanced settings** page, provide information about the infrastructure you already provisioned, such as settings for the VPC, DB subnet group, and security group. In addition, you can provide custom options for encryption, backup retention period, maintenance window, and so on. You will also create a user to administer this database.
7. For the database name, you can provide the name you want to use for the CMS database. You can also change the database port from the default value to your choice of port.
8. Choose **Create database**, and then wait for the DB instance status to change to **available** in the Amazon RDS console.
9. Choose the **Instances** view and note the **Endpoint** name. In case of failover to another Availability Zone, this endpoint enables an application to reconnect to a new primary database instance without having to change anything.
10. (Optional) Create a CNAME in Route 53 or other DNS server for the database cluster endpoint. Use this CNAME during the installation of SAP BOBI Platform nodes.

Step 7. Install CMS Database (Only for CMS Database on EC2 Instance)

Install the CMS database with an SAP supported database version of your choice. Refer to the database vendor specific documentation for instructions. You can also install Audit database if you plan to use auditing. The Auditing database can be installed at a later point in time as it is not required for SAP BOBI Platform functioning.

Step 8. Install SAP BOBI Platform Nodes

1. Log in to each EC2 instance in the SAP BOBI Platform server and repeat the following step to install SAP BOBI platform on each instance.
2. See the [SAP BusinessObjects BI Platform installation guide](#) and go to the SAP BOBI documentation specific to the version you want to install. Launch the installation as described:

Custom / Expand > Expand an existing SAP BusinessObjects BI platform deployment > Instances > Servers > Platform Services
3. For the first server installation, choose **Start a new SAP BusinessObjects BI platform deployment**. Follow the instructions and enter inputs as required for example database

connection information. Figure 3 shows example of adding database connection information when using RDS MySQL.

4. (Optional) This step is only required for multi-node installation. For all additional server installations, choose **Expand an existing SAP BusinessObjects BI platform deployment**. Follow the instructions and enter inputs as required for example database connection information and first CMS server connection information.

Figure 3 is an example input for CMS database information for Linux installation. In this case, RDS MySQL database is used on default port 3306.

```
Configure CMS Repository Database - MySQL
Enter connection information about the existing database to use
as the CMS repository

CMS Database Name
[boedb ]
MySQL Server
[mybobjdb.abcdabcdadb.us-west-2.rds.amazonaws.com ]
MySQL Port
[3306 ]
User Name
[admin ]
Password
[***** ]
Reset existing database (1 = yes, 0 = no)[1]

Press Tab to go to the next field, Ctrl+B to go the previous field, Ctrl+X to quit, or Enter to continue.
```

Figure 3: Example of adding database connection information when using RDS MySQL

This completes the installation of SAP BOBI Platform.

Step 9. Configure End User Access for Multi-Node Deployment

To distribute the user load evenly across the web tier servers, you can use a load balancer between the web users and the web servers. In this guide, we'll discuss the use of [Elastic Load Balancing \(ELB\)](#) for this purpose. You can also install other load balancers on EC2 instances for end user access, refer to vendor specific documentation for such installation. An Application Load Balancer automatically scales its request handling capacity in response to incoming application traffic. Follow these steps to configure an Application Load Balancer for SAP BOBI Platform:

1. In the [Amazon EC2 console](#), [create an Application Load Balancer](#) in the VPC where SAP BOBI Platform is running. Specify the Availability Zones and subnets of all the web tier servers.

 **Note**

Application Load Balancer cannot route fields with special characters (such as, underscore) to targets. Disable the `routing.http.drop_invalid_header_fields` attribute to enable routing of fields with special characters.

2. [Configure a security group](#) that allows users to connect to the Application Load Balancer on the SSL port.
3. [Create a target group](#) to register web servers as the targets to the load balancer. For **Target type**, choose **ip** and specify the IP address and SSL port of the web servers to register as targets.
4. [Enable sticky sessions](#).
5. Create or upload an existing SSL certificate in AWS Certificate Manager (ACM).
6. Enable Secure Sockets Layer (SSL) communications for SAP BOBI Platform by following the instructions in the [Business Intelligence Platform Administrator Guide](#). See also: [Enabling SSL in BI Platform 4.2 SP05](#) on the SAP Blog.
7. (Optional) Create a CNAME in Amazon Route 53 for the Application Load Balancer DNS name. Use this CNAME to access SAP BOBI Platform.

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 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 in conjunction with a special tag filter. See [AWS Tagging Strategies](#) and use it as a starting point to define the tags you need for your resources. Below are the examples on how you can use tags for operational needs:

- 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. You can use services like [Amazon CloudWatch](#) and [AWS CloudTrail](#) to monitor your underlying infrastructure and APIs, respectively. CloudWatch provides ready-to-use KPIs for CPU and 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.

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 SUSE Manager or Red Hat CloudForms)
- third-party products such as those available on AWS Marketplace
- using AWS Systems Manager

Here we outline some key operating system maintenance tasks.

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 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 fulfill your

requirements, there are third-party products available as well. Some of these are available via 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.

Administrator Access

You can access the backend SAP systems for administration purposes in a number of ways:

- AWS Systems Manager Session Manager
- SSH

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 select the **No Reboot** option. You can use the AWS Systems Manager Run Command to take [application-consistent snapshots of all EBS volumes](#) attached to your instance.

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, 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 more information, see [Amazon AMIs](#).

Syntax:

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

File Backup to S3

You can perform traditional file-based backups to Amazon S3 from your EBS volumes. 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 a number that have been certified by SAP. For more information, see [AWS SAP Partner Solutions](#).

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 EFS filesystem 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 then be automated and scheduled using AWS Systems Manager Run Command in combination with Amazon CloudWatch Events.

Backing up SAP BOBI Platform

Backup of SAP BOBI should protect the following components. The backup of CMS database and FileStore should be taken at the same time to maintain consistency.

- CMS Database (Amazon RDS or Database on EC2)
- FileStore (Amazon EFS for multi-node install or Amazon EBS for standalone install)

- SAP BOBI installation directory

You can choose from following options for backup.

- When using Amazon RDS for CMS database and Linux operating system for application, you can use [AWS Backup](#) as a central tool for backups. AWS Backup is a fully managed backup service that makes it easy to centralize and automate the back up of data across AWS services in the cloud. You can configure backup policies based on tags from a central backup console, simplifying backup management and making it easy to ensure that your application data is backed up and protected. You can put database, FileStore, and installation directory resources in same policy to ensure consistency.
- You can use supported third-party backup tools that provides database and file system agents for backup and recovery of all SAP BOBI platform component.
- If the preceding AWS services and tools do not meet your requirements, you can also use standard database backup tools and scripts to create database backups, file system backups and EBS snapshots. Database and file system backups can be stored on an EBS volume attached to your database and application EC2 instances. For better durability and agility, we recommend that you move your backups to Amazon S3. Based on your business continuity and compliance requirements, you can choose to move your backups to Amazon S3 Glacier and use Amazon S3 Lifecycle policies. For details, see [How Do I Create a Lifecycle Policy for an S3 Bucket?](#). Amazon S3 Lifecycle policies also let you delete older backups based on your backup retention requirements.
- If you use SAP HANA as the database for CMS, you can use AWS Backint Agent for SAP HANA to backup your data to Amazon S3. For more information, see [AWS Backint Agent for SAP HANA](#).

Recovering the SAP BOBI Platform

The backups that you choose for the restore of CMS database and FileStore should have been created at the same time to maintain consistency. You can recover a database to a point-in-time using log files, but Amazon EFS (used for FileStore) does not have similar capabilities. In this case, recovering a database to most recent state but FileStore to an older state may cause inconsistencies between the two.

Based on the backup strategy, the following are options for restore.

- When you restore a backup in AWS Backup, a new resource is created based on the backup that you are restoring. Depending on the component that you restore, you can point you SAP BOBI

Platform installation to the new resource or copy data to the original resource. For example, you can restore Amazon EFS and Amazon EBS on a different file system or EBS volume. After you have the new resource available, you can either copy a subset of the data or replace your original resource with the new one. See [Restoring a Backup](#) for details.

- When restoring using third-party software, refer to vendor- and application-specific documentation.
- If you are restoring from Amazon S3 using custom scripts, you will have to restore the backup to an EBS volume, and then use either database specific tools or native operating system features to restore your data back to SAP BOBI Platform installation.

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, NVMe device IDs associated with the volume could change. To avoid mount errors during changes of instance type or instance reboots, you need to create a label for your file systems and mount it by label than the actual NVME ids.

Aside from operating system maintenance, there is also maintenance you can consider for EC2 instances themselves. This can be driven via AWS Systems Manager Automation documents. Some examples of this 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 on a schedule by using Amazon CloudWatch Events or by using 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 by 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.

Finally, use the [AWS Instance Scheduler](#) Solution to easily configure custom start and stop schedules for their Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Relational Database Service (Amazon RDS) instances.

Cost Optimization

Just as with right sizing, we recommend customers make cost optimization an ongoing process. This is an extensive topic with many Use the **AWS-StopEC2InstanceWithApproval** document services that help with budgeting, cost control and proactive cost optimization recommendations.

For more details, see the [Cost Optimization Pillar](#) of the AWS Well-Architected Framework and the [SAP on AWS Pricing and Optimization Guide](#).

Automation

Automation using Infrastructure as Code with AWS CloudFormation

We recommend following the principle of Infrastructure as code (IaC) in 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, and thus follow the principle of IaC.

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 Linux already available.

Integration with AWS Big Data Services

The SAP BOBI Platform product can use multiple AWS Big Data services as data sources for reporting purposes. When using SAP BOBI version 4.2, you can connect to the following AWS data sources:

Table 2: AWS Big Data services support for SAP BusinessObjects Business Intelligence 4.2

Use Case	Amazon Product	SAP BOBI 4.2 Supported
Data source	Amazon RDS Oracle	Yes
Data source	Amazon Redshift	Yes
Data source	Amazon EMR Hive (Hive1)	Yes
Data source	Amazon EMR Hive (Hive2)	Yes
Data source	Amazon EMR Hive 5.6 (Hive2)	Yes

See the [SAP Product Availability Matrix \(PAM\)](#) for the complete list of SAP BOBI Platform supported data sources specific to your version.

Support

To get help from SAP, SAP and AWS requires a business support agreement on AWS. [AWS Business Support](#) provides resources and technical support for customers running SAP workloads on AWS. If you have any technical issues around AWS, you can open up a case with SAP or AWS and it will be routed to the appropriate team. AWS also offers [AWS Enterprise Support](#) for customers running mission critical production workloads on AWS.

Document Revisions

Date	Change
January 2023	Updates throughout the guide
October 2019	Initial publication

SAP BusinessObjects Business Intelligence Platform on AWS Deployment and Operations Guide for Windows

SAP specialists, Amazon Web Services

Last updated: January 2023

The purpose of this guide is to provide an overview of how to implement and operate SAP BusinessObjects (BO) Business Intelligence (BI) Platform (also referred in this document as SAP BOBI Platform) on Amazon Elastic Compute Cloud (Amazon EC2). This guide covers common AWS services and features that are relevant for SAP BusinessObjects BI platform. This guide is not an exhaustive list of all possible configuration options. It covers solutions common to typical deployment scenarios.

This guide is not intended to replace the standard SAP BOBI Platform installation and administration guides, operating system, or relational database management system (RDBMS) documentation.

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](#).

The purpose of this guide is to provide an overview of how to implement and operate SAP BusinessObjects (BO) Business Intelligence (BI) Platform (also referred in this document as SAP BOBI Platform) on Amazon Elastic Compute Cloud (Amazon EC2). This guide covers common AWS services and features that are relevant for SAP BusinessObjects BI platform. This guide is not an exhaustive list of all possible configuration options. It covers solutions common to typical deployment scenarios.

This guide is not intended to replace the standard SAP BOBI Platform installation and administration guides, operating system, or relational database management system (RDBMS) documentation.

General AWS 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, see [Getting Started with AWS](#).)

- [Amazon EC2](#)
- [Amazon VPC](#)
- [AWS Identity and Access Management \(IAM\)](#)
- [Amazon EBS](#)
- [Amazon S3](#)
- [AWS Systems Manager](#)
- [AWS CloudFormation](#)

Prerequisites

Before you start implementing your SAP BOBI Platform systems, we recommend that you review these prerequisites to ensure there are minimal interruptions and delays.

Recommended Reading

We also recommend you first read some key overview and best practice guides:

- [SAP on AWS Overview and Planning Guide](#)
- [Getting Started with Architecting SAP on the AWS Cloud](#)
- [Best Practices for Windows on Amazon EC2](#)

SAP Notes listed in the following table have useful information regarding SAP BOBI deployment in AWS.

SAP Notes for SAP BOBI deployment on AWS

SAP Note	Description
1588667	SAP on AWS: Overview of related SAP notes and web links

SAP Note	Description
1656099	SAP on AWS: Supported products, platforms, and landscapes
2442979	Amazon S3 recommendations for SAP BusinessObjects Business Intelligence Platform
2438592	BI Platform 4.2 Cloud Support

Technical Requirements

- Ensure that any services you will use for your SAP BOBI Platform deployment are not constrained by default AWS service limits. You can find the details at [AWS Service Limits](#). You can increase soft limits by submitting a support ticket to AWS.
- Make sure that the following information is available in relevance to your existing AWS resources. You will need this information while executing AWS Command Line Interface (AWS CLI) commands to create your Amazon EC2 and Amazon Elastic Block Store (Amazon EBS) resources:

Information You Need	Description
Region ID	AWS Region where you want to deploy your AWS resources.
Availability Zone	Availability Zone within your target region where you want to deploy your resources.
Amazon VPC ID	Amazon Virtual Private Cloud (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.

Information You Need	Description
	You can find latest Windows 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 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 CLI tools.

- Ensure that you have a key pair that you can use to launch your Amazon EC2 instances. See [Amazon EC2 Key Pairs and Windows Instances](#) if you need to create a key.
- Ensure that you have the network details like VPC ID, Subnet ID, and so on, of the VPC where you plan to launch your Amazon EC2 instances to host your SAP BOBI Platform applications.
- Ensure that the required ports are open on the security group attached to your Amazon EC2 instance to allow log in to the operating system.
- For distributed or high availability (HA) installations, ensure that the security group attached to each application servers allows communication over the required ports between them. The easiest way to do this is to create a rule that references a security group as its own source and allow traffic on the required ports for that rule.
- If you intend to use the AWS CLI to launch your instances, then ensure that you have installed and configured AWS CLI with the appropriate credentials. See [Configuring the AWS CLI](#) for more details.
- If you intend to use the AWS Management Console to launch your instances, then ensure that your IAM user has permission to launch and configure Amazon EC2, Amazon EBS, and so on. See the [IAM documentation](#) for more details.
- Ensure that you have the required SAP software available either via an Amazon Simple Storage Service (Amazon S3) bucket or on a file share accessible from an Amazon EC2 instance. If you use

Amazon S3, make sure to assign appropriate IAM role permissions to the EC2 instance to allow S3 access.

- All enterprise customers use DNS service. You can create a hosted zone in Amazon Route 53. You can optionally use AWS Directory Service for Microsoft Active Directory, also known as AWS Managed Microsoft AD. This service lets your directory-aware workloads and AWS resources use managed Active Directory in the AWS Cloud. For more details on this service, see [AWS Directory Service](#) and [Create Your AWS Managed Microsoft AD directory](#).

Planning

The following topics are important for planning the SAP BOBI Platform in AWS Cloud.

Topics

- [Choosing a Region](#)
- [Choosing an Availability Zone](#)
- [Architecture Options](#)
- [Storage](#)
- [Sizing](#)
- [High Availability \(HA\) and Disaster Recovery \(DR\)](#)
- [Security & Compliance](#)
- [Operating System](#)
- [Compute](#)
- [Network](#)
- [Storage Services](#)

Choosing a Region

When choosing which AWS Region to deploy your SAP environment in you should consider the following topics:

- Proximity to your on-premises data centers, systems, and end users to minimize network latency.
- Data residency and compliance requirements.

- Whether the AWS products and services you plan to use are available in the Region. For a detailed list of AWS products and services by Region, see the [Region Table](#) page on the AWS website.
- The Amazon EC2 instances types you plan to use are available in the Region. To view AWS Region availability for a specific instance type, see the [Amazon EC2 Instance Types for SAP](#) page.

Choosing an Availability Zone

No special considerations are required when choosing an Availability Zone for your SAP deployment on AWS. If high availability (HA) is a requirement, use multiple Availability Zones. For more information about HA, see the [SAP BusinessObjects BI Platform on AWS: HA/DR Guide for Windows](#).

See [AWS Global Infrastructure](#) for information about AWS Regions and Availability Zones.

Architecture Options

The server-side architecture of SAP BOBI Platform consists of five tiers: web, management, storage, processing, and data. (For details, see the administrator's guide on the [SAP BusinessObjects Business Intelligence Platform](#) website). The following list provides high-level details.

- **Management tier:** Includes the CMS servers, event servers, and associated services.
- **Storage tier:** Includes input and output file repository servers. The file system used by these servers to store files, such as documents, reports, and universes, must be on a shared file system.
- **Web tier and processing tier:** Performs functions like receiving and processing user requests.
- **Data tier:** Consists of the CMS system database and the auditing data store.

You can have following example architecture designs for the above tiers:

- Install all tiers on the same EC2 instance.
- Install the application and database tiers on two separate EC2 instances.
- Install different tiers on multiple EC2 instances grouped based on customer-specific requirements.

The architecture choice depends on multiple factors like complexity, cost, sizing, and technical restrictions. For example, if you use [Amazon RDS](#) as the database, application tiers cannot be installed with database.

CMS and Audit Database Architecture Options

You have the choice of deploying the SAP BOBI Platform application on a standard SAP supported database like SAP HANA, SAP ASE, IBM DB2, Microsoft SQL Server, or [Amazon Relational Database Service \(Amazon RDS\)](#). For supported [Amazon RDS](#) database types, see [SAP Note 1656099 SAP on AWS: Supported SAP, DB/OS and Amazon EC2 products](#).

[Amazon RDS](#) is a service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. Amazon RDS takes over many of the difficult or tedious management tasks such as backups, software patching, automatic failure detection, and recovery. You can read more about this service in [Amazon RDS documentation](#).

Figure 1 shows an example large scale architecture of SAP BOBI with multi-AZ and multi-instance architecture. Web, Management, Processing, and Data tiers are all distributed on different EC2 instances. [Amazon RDS](#) MySQL is used for CMS database.

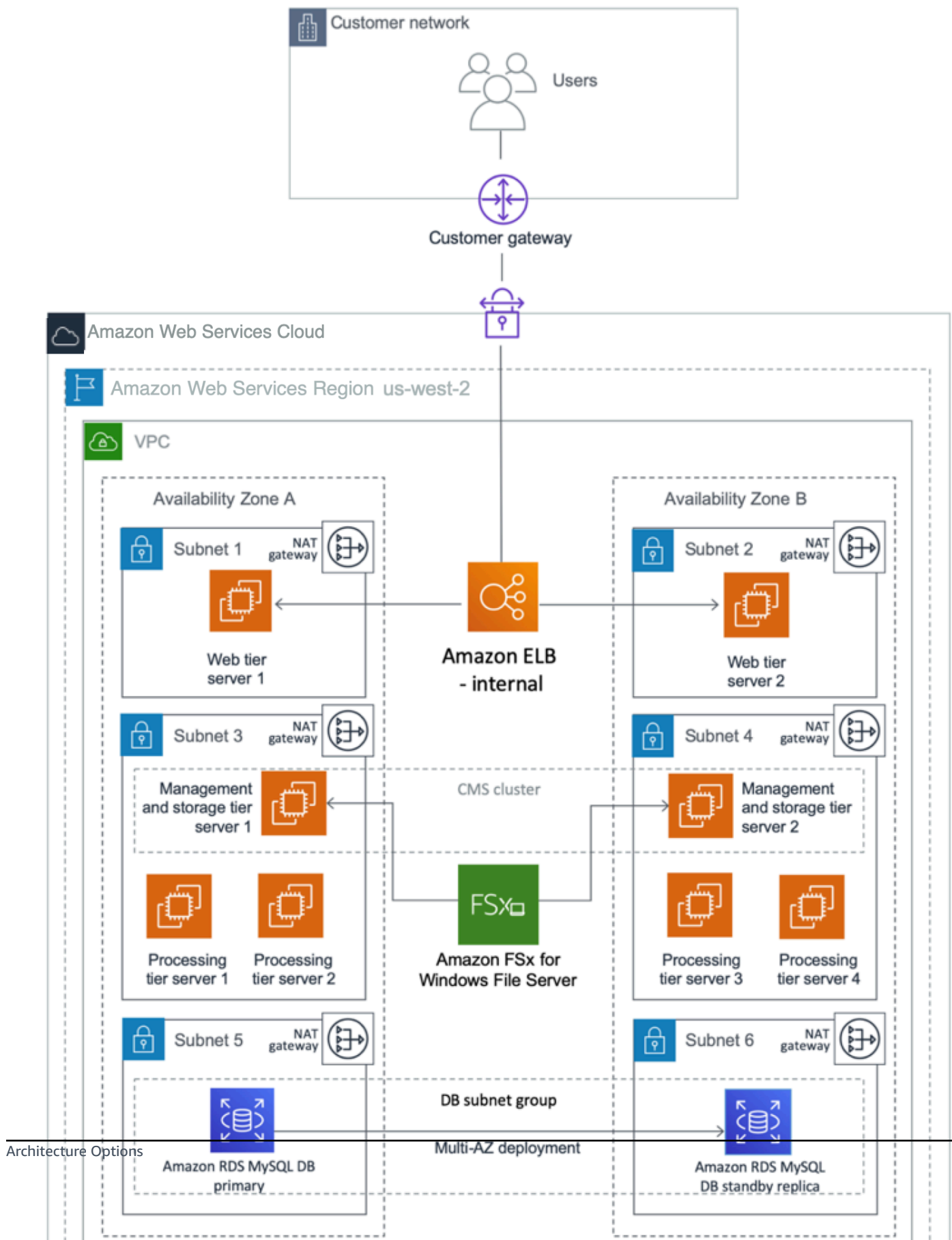


Figure 1: SAP BOBI with multi-AZ and multi-instance architecture

Storage

See the [Sizing](#) section for resources on SAP's standard recommendations. If no storage performance requirements are available, AWS recommends General Purpose SSD (gp3) as the default EBS volume type for SAP workloads.

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 will use standard Windows filesharing features to share these directories from the EC2 instance hosting the central services. The sapinst.exe installer will create 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 such as those available from the [AWS Marketplace](#) or custom-built solutions), but that is beyond the scope of this guide. If using such a solution in the context of a high-availability installation, consider that the NFS solution could itself be a single point of failure without appropriate protection.

Sizing

At a high level, BOBI platform sizing is a two-step process. The first step is to get SAPS through the SAP Sizing tool [Quick Sizer](#). The second step is to map the output to appropriate Amazon EC2 instance types. See the SAP BOBI sizing guide available from the [SAP BusinessObjects Business Intelligence Platform help documentation](#). Follow the standard SAP Quick Sizer to determine the right SAPS requirements for your workload. Once you have identified the SAPS numbers, you can use any of the AWS [SAP certified instances](#) for your SAP BOBI Platform instances. Make sure to read the SAP BusinessObjects Business Intelligence section of the note carefully to ensure that you meet the version, EC2 resource, operating system, and database requirements for AWS.

For sizing your storage requirements for the database and app tier, AWS provides various volume types, such as general purpose SSDs (gp3) and provisioned IOPS SSD (io2). You can read more about the differences at [Amazon EBS Volume Types](#). As general guidance, we recommend that you consider the gp3 volume type as a starting point to see if it satisfies your throughput and IOPS requirement of your workload, and consider io2 volume types if you have a need for sustained IOPS.

It is easy to add capacity to your existing SAP BOBI deployment in AWS. Therefore, for migrating your existing deployment to AWS, you may not need the same degree of over-provisioning as is typical for on-premises environments.

High Availability (HA) and Disaster Recovery (DR)

If you require a highly available BOBI environment, then it critical to design the HA and DR environment that can support the recovery time objective (RTO) and recovery point objective (RPO) that your business teams have established. For more information, see the [SAP BusinessObjects BI Platform on AWS: HA/DR Guide for Windows](#).

Security & Compliance

The following AWS security resources help you achieve the level of security you require for your SAP NetWeaver environment on AWS:

- [AWS Cloud Security](#)
- [CIS AWS Foundations Benchmark](#)
- [Introduction to AWS Security](#)
- [Best Practices for Security, Identity, & Compliance](#)
- [AWS Well-Architected Framework Security Pillar](#)
- [Network and security features for Windows and Amazon EC2](#)

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](#)
- Use [Amazon Inspector](#), an automated security assessment service that helps you test the network accessibility of your Amazon EC2 instances and the security state of your applications running on the instances.

Encryption

Security is a priority on AWS. 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 in transit 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 they require no additional action from you or your applications.

Similarly, all Amazon FSx file systems are encrypted at rest with keys 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 you don't have to modify your applications.

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

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 you scale workloads horizontally or require high availability, you may choose to include multiple, functionally similar, EC2 instances in the same security group. In this case, you'll need to add a rule to your security groups.

Some configuration changes may be necessary in the security groups, route tables, and network ACLs. You can refer to the operating system product documentation, or other sources such as the [Security Group Rules Reference](#) in the Amazon Elastic Compute Cloud (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 may set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

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 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 AWS 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 produced by CloudTrail enables 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 set up notifications on SSH login to your email address or mobile phone.

Operating System

If you plan on using Windows other than via Amazon EC2 for Windows Server, then ensure you have the appropriate licenses in place and the appropriate tenancy type selected. For more details, refer to your licensing terms and conditions, and see [Windows on AWS](#).

A base AMI is required to launch an Amazon EC2 instance. For SAP workloads on Windows, you must have a minimum of Windows Server 2012 R2 to be supported as previous versions of Windows Server are now out of support by SAP. If you are using BYOL instead of license-included for Windows Server, you must create your own AMI. For details, see the [Windows on AWS licensing documentation](#). Ensure that you have access to the appropriate Windows Server AMIs before proceeding further.

As with any operating system, we recommend you keep the OS up-to-date with the latest patches. You can also refer to [SAP Note 2325651: Required Windows Patches for SAP Operations](#).

Compute

AWS has certified multiple instance families with different sizes to run SAP workloads. The details of the latest list of EC2 instance types certified are all specified in our webpage [Amazon EC2 Instance Types for SAP](#).

Select the appropriate EC2 instance type based on your requirements as per the sizing section. AWS recommends you use the latest generation of your selected instance family where possible (assuming it is SAP certified and therefore listed in the above webpage).

Network

Ensure that you have your network constructs set up to deploy resources related to your SAP workload. If you haven't already set up network components like Amazon Virtual Private Cloud (Amazon VPC), subnets, route tables and so on., you can use the [AWS Quick Start for Modular and Scalable VPC Architecture](#) to easily deploy scalable VPC architecture in minutes. Refer to the deployment guide for more details, then set up your EC2 instances for the SAP workload within this VPC.

You must also set up a secured network connection between the corporate datacenter and the VPC, along with appropriate route table configuration, if this is not already configured.

Storage Services

The SAP BOBI Platform uses the following AWS storage services:

- [Amazon Elastic Block Store \(Amazon EBS\)](#) is used for block storage requirements of SAP BOBI Platform application servers and databases (when the database is installed on EC2).

Figure 2 shows an example use of EBS volumes for application and database. In this example, EBS volumes are used for root volumes, SAP BOBI Platform installation directory, operating system swap volume, and database data and log volumes. The CMS database is typically a small database that stores information like users, SAP BOBI Platform servers, folders, and other configurations. Therefore, it does not have the same storage performance requirements as other enterprise OLTP/OLAP databases. Follow the best practices of the database vendor for designing storage for the SAP BOBI Platform database.

- [Amazon FSx for Windows File Server](#) is used for shared file system requirements of SAP BOBI Platform application servers installed on Windows EC2 instances. The usage is the same as for FileStore as described above.
- [Amazon Simple Storage Service \(Amazon S3\)](#) is used for storing the backups of SAP BOBI Platform application servers.

Figure 2 shows an example use of AWS storage services by an SAP BOBI Platform installation. In this example, two SAP applications servers and a database are installed on three separate EC2 instances with Windows operating systems. EBS volumes are used for local file systems like root, install, swap, data, and log volumes. [Amazon FSx for Windows File Server](#) is used for shared file system FileStore.

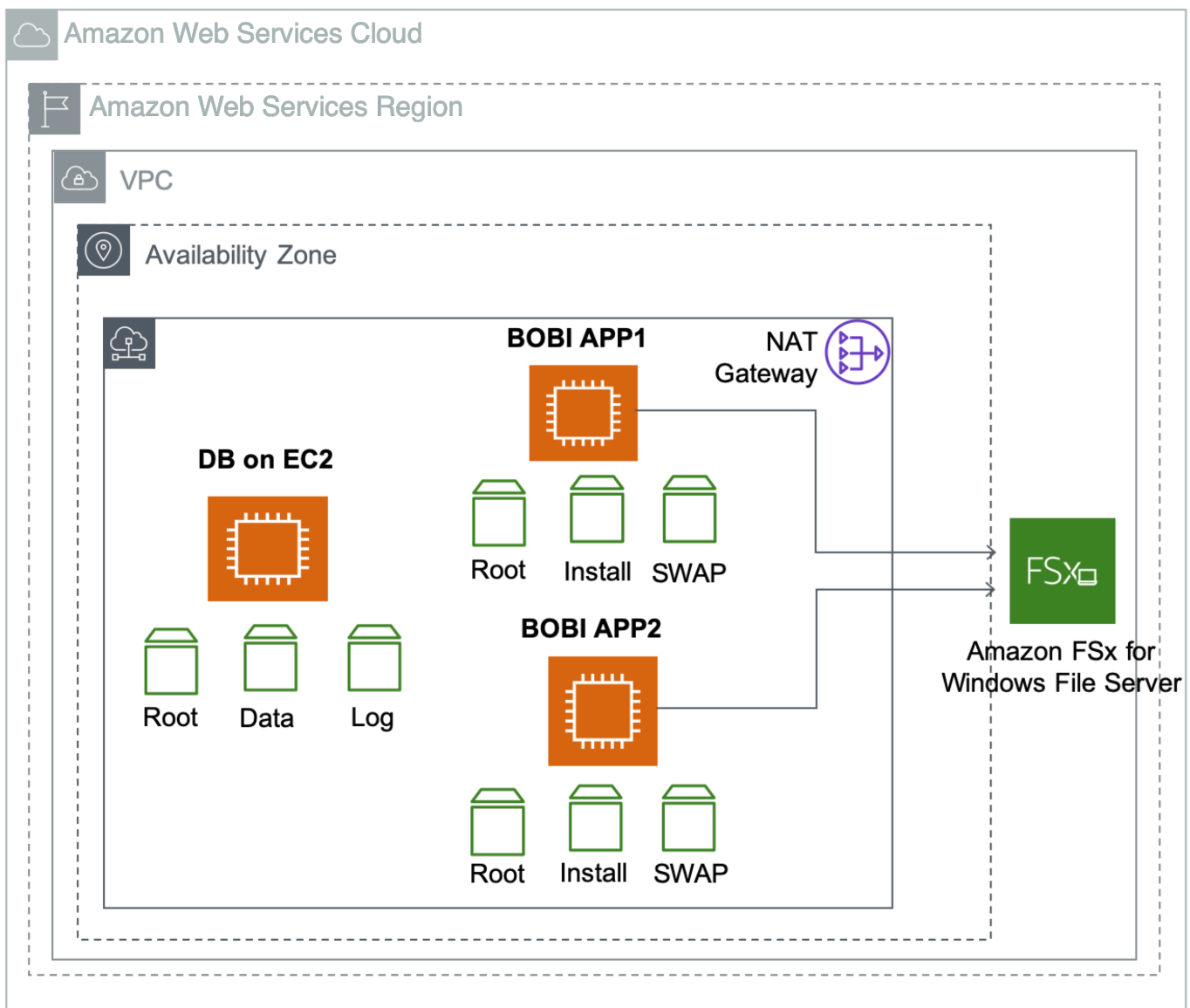


Figure 2: AWS storage system use on SAP BOBI Platform installation

Deployment

In this deployment, we will provision an Amazon EC2 instance for installing the SAP application servers and the CMS database (if you are using database on EC2). When using Amazon RDS for CMS database, follow [Step 7. \(Only for CMS Database on EC2 Instance\) Installing CMS Database](#).

In this deployment, we will provision an Amazon EC2 instance for installing a standalone Oracle database standard system.

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 see the documentation.

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 our scenario, we are using the AWS Launch Wizard for SAP in a single-AZ deployment to create the VPC, subnets, security groups, and IAM roles. This is just an example setup and customers should follow their own network layout and comply with their own security standards. This may include:

- using an AWS Launch Wizard for SAP for a multi-AZ deployment of SAP HANA
- using a landing zone solution like [AWS Control Tower](#)
- work with their cloud team (for example a Cloud Center of Excellence or CCoE) to use existing standards

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

Note

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

2. 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 the SAP workload.

Step 2. Create a JSON file for the Amazon EBS storage

Create a JSON file that contains the storage requirements for SAP BOBI Platform server volumes.

Below is an example JSON file with two EBS volumes for swap and SAP BOBI Platform installation directory. You can modify this file as per your requirements:

```
[
  {
    "DeviceName": "/dev/sdh",
    "Ebs": {
      "VolumeSize": 32,
      "VolumeType": "gp3",
      "DeleteOnTermination": true
    }
  },
  {
    "DeviceName": "/dev/sdg",
    "Ebs": {
      "VolumeSize": 50,
      "VolumeType": "gp3",
      "DeleteOnTermination": true
    }
  }
]
```

Step 3. Launch the Amazon EC2 Instance

Launch the Amazon EC2 instance for the SAP BOBI Platform 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.

```
$ 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. Create a JSON file for the Amazon EBS storage](#).

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-xxxxxxxxxxxxxxxx --count 1 \  
--instance-type m5.large --key-name=my_key --security-group-ids \  
sg-xxxxxxx --subnet-id subnet-xxxxxx \  
--block-device-mappings file://C:\Users\<file>.json
```

You can also launch EC2 instances using the AWS Management Console. For detailed steps, see [Launch Windows EC2 Instances using AWS Management Console](#).

Step 4. 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 the new [Session Manager feature for AWS Systems Manager](#) (for command line tasks), or by following these steps:
 - a. In the AWS Management Console, select the EC2 instance **jumpbox** and choose **Connect**. Download the RDP file from the pop-up that appears.
 - b. Choose **Get Password** and provide your private key to decrypt the password. This is the password for the local Administrator user on **jumpbox**.
 - c. Open the RDP file in your preferred RDP program, and connect to **jumpbox**. Log in with user Administrator and the password you just retrieved in Step 2.
 - d. Once logged in, return to the AWS Management Console and repeat steps 1 and 2, but this time for the EC2 instance where you will install SAP. Copy the downloaded RDP file to **jumpbox**.
 - e. While logged into **jumpbox**, open the RDP file for the SAP instance 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 version of the Java JRE that is compatible with your desired SAP installation software.
5. Install the AWS Data Provider as per the instructions for Windows in the [AWS Data Provider for SAP Installation and Operations Guide](#).
6. Install and configure the AWS Systems Manager Agent (SSM Agent). For steps, see [Working with SSM Agent](#).

Step 5. Create Amazon FSx Volumes

1. The global file share and transport directories need to be available across all your SAP system's EC2 instances. For Windows we assume use of Amazon FSx for this purpose.
2. Be sure 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 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 6. Prepare and Install the CMS Database (Only for RDS Database)

This option is applicable only when Amazon RDS MySQL is used for the CMS database. You can create a separate database for the auditing database if it's required.

1. Create a DB subnet group for an RDS instance by following the instructions in [Create a DB Subnet Group](#).
2. In the [Amazon RDS console](#), launch an Amazon RDS MySQL DB instance by following the instructions in the [Creating a DB Instance Running the MySQL Database Engine](#).
3. Choose a supported DB version based on [SAP Note 1656099 - SAP on AWS: Supported SAP, DB/OS and Amazon EC2 products](#), and select the instance type and storage based on your sizing output.

4. On the **Specify DB details** page, in the **Instance specifications** section, choose **Create replica in different zone**.
5. The **Choose use case** page asks if you are planning to use the DB instance you are creating for production. If you choose **Production - MySQL**, the Multi-AZ failover option is preselected. You can deselect this option if you are not installing a highly available system.
6. On the **Configure advanced settings** page, provide information about the infrastructure you already provisioned, such as settings for the VPC, DB subnet group, and security group. In addition, you can provide custom options for encryption, backup retention period, maintenance window, and so on. You will also create a user to administer this database.
7. For the database name, you can provide the name you want to use for the CMS database. You can also change the database port from the default value to your choice of port.
8. Choose **Create database**, and then wait for the DB instance status to change to **available** in the Amazon RDS console.
9. Choose the **Instances** view and note the **Endpoint** name. In case of failover to another Availability Zone, this endpoint enables an application to reconnect to a new primary database instance without having to change anything.
10. (Optional) Create a CNAME in Amazon Route 53 or other DNS server for the database cluster endpoint. Use this CNAME during the installation of SAP BOBI Platform nodes.

Step 7. Install CMS Database (Only for CMS Database on EC2 Instance)

Install the CMS database with an SAP supported database version of your choice. Refer to the database vendor specific documentation for instructions. You can also install Audit database if you plan to use auditing. The Auditing database can be installed at a later point in time as it is not required for SAP BOBI Platform functioning.

Step 8. Install SAP BOBI Platform Nodes

1. Log in to each EC2 instance in the SAP BOBI Platform server and repeat the following step to install SAP BOBI platform on each instance.
2. See the [SAP BusinessObjects BI Platform installation guide](#) and go to the SAP BOBI documentation specific to the version you want to install. Launch the installation as described:

Custom / Expand > Expand an existing SAP BusinessObjects BI platform deployment > Instances > Servers > Platform Services

3. For the first server installation, choose **Start a new SAP BusinessObjects BI platform deployment**. Follow the instructions and enter inputs as required for example database connection information. Figure 3 shows example of adding database connection information when using RDS MySQL.
4. (Optional) This step is only required for multi-node installation. For all additional server installations, choose **Expand an existing SAP BusinessObjects BI platform deployment**. Follow the instructions and enter inputs as required for example database connection information and first CMS server connection information.

This completes the installation of SAP BOBI Platform.

Step 9. Configure End User Access for Multi-Node Deployment

To distribute the user load evenly across the web tier servers, you can use a load balancer between the web users and the web servers. In this guide, we'll discuss the use of [Elastic Load Balancing \(ELB\)](#) for this purpose. You can also install other load balancers on EC2 instances for end user access, refer to vendor specific documentation for such installation. An Application Load Balancer automatically scales its request handling capacity in response to incoming application traffic. Follow these steps to configure an Application Load Balancer for SAP BOBI Platform:

1. In the [Amazon EC2 console](#), [create an Application Load Balancer](#) in the VPC where SAP BOBI Platform is running. Specify the Availability Zones and subnets of all the web tier servers.

Note

Application Load Balancer cannot route fields with special characters (such as, underscore) to targets. Disable the `routing.http.drop_invalid_header_fields` attribute to enable routing of fields with special characters.

2. [Configure a security group](#) that allows users to connect to the Application Load Balancer on the SSL port.
3. [Create a target group](#) to register web servers as the targets to the load balancer. For **Target type**, choose **ip** and specify the IP address and SSL port of the web servers to register as targets.
4. [Enable sticky sessions](#).
5. Create or upload an existing SSL certificate in AWS Certificate Manager (ACM).

6. Enable Secure Sockets Layer (SSL) communications for SAP BOBI Platform by following the instructions in the [Business Intelligence Platform Administrator Guide](#). See also: [Enabling SSL in BI Platform 4.2 SP05](#) on the SAP Blog.
7. (Optional) Create a CNAME in Amazon Route 53 for the Application Load Balancer DNS name. Use this CNAME to access SAP BOBI Platform.

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 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 in conjunction with a special tag filter. See [AWS Tagging Strategies](#) and use it as a starting point to define the tags you need for your resources. Below are the examples on how you can use tags for operational needs:

- 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. You can use services like [Amazon CloudWatch](#) and [AWS CloudTrail](#) to monitor your underlying infrastructure and APIs, respectively. CloudWatch provides ready-to-use KPIs for CPU and 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.

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

- third-party products such as those available on AWS Marketplace
- using AWS Systems Manager

Here we outline some key operating system maintenance tasks.

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 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 fulfill your requirements, there are third-party products available as well. Some of these are available via 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.

Administrator Access

You can access the backend SAP systems for administration purposes in a number of ways:

- AWS Systems Manager Session Manager
- Remote Desktop Protocol
- SSH

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 select the **No Reboot** option. You can use the AWS Systems Manager Run Command to take [application-consistent snapshots of all EBS volumes](#) attached to your instance using Windows Volume Shadow Copy Service (VSS) to make it safe to create the image without rebooting the instance.

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, 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 more information, see [Creating an Amazon EBS Backed Windows AMI](#).

Syntax:

```
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 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 S3

You can perform traditional file-based backups to Amazon S3 from your EBS volumes. 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 a number 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 Simple Storage Service (Amazon S3). Amazon FSx backups are incremental, which means that only the changes 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 was established when you created the file system.

If you want to set up a custom backup schedule, you can [deploy our reference solution](#).

Backing up SAP BOBI Platform

Backup of SAP BOBI should protect the following components. The backup of CMS database and FileStore should be taken at the same time to maintain consistency.

- CMS Database (Amazon RDS or Database on EC2)
- FileStore (Amazon FSx for multi-node install or Amazon EBS for standalone install)

- SAP BOBI installation directory

You can choose from following options for backup.

- For Amazon FSx, you can schedule the backups. For details, see [Working with Backups](#).
- When using Amazon RDS for CMS database and Windows operating system for application, you can use AWS Backup for Amazon EBS and database backups. AWS Backup is a fully managed backup service that makes it easy to centralize and automate the back up of data across AWS services in the cloud. You can configure backup policies based on tags from a central backup console, simplifying backup management and making it easy to ensure that your application data is backed up and protected. You can put database, FileStore, and installation directory resources in same policy to ensure consistency.
- When using Amazon RDS for CMS database and Windows operating system for application, you can use [AWS Backup](#) service for the backup of Amazon RDS and SAP BOBI Installation directory.
- You can use supported third-party backup tools that provide database and file system agents for backup and recovery of all SAP BOBI platform component.
- If the preceding AWS services and tools do not meet your requirements, you can also use standard database backup tools and scripts to create database backups, file system backups and EBS snapshots. Database and file system backups can be stored on an EBS volume attached to your database and application EC2 instances. For better durability and agility, we recommend that you move your backups to Amazon S3. Based on your business continuity and compliance requirements, you can choose to move your backups to Amazon S3 Glacier and use Amazon S3 Lifecycle policies. For details, see [How Do I Create a Lifecycle Policy for an S3 Bucket?](#). Amazon S3 Lifecycle policies also let you delete older backups based on your backup retention requirements.
- If you use SAP HANA as the database for CMS, you can use AWS Backint Agent for SAP HANA to backup your data to Amazon S3. For more information, see [AWS Backint Agent for SAP HANA](#).

Recovering the SAP BOBI Platform

The backups that you choose for the restore of CMS database and FileStore should have been created at the same time to maintain consistency. You can recover a database to a point-in-time using log files, but Amazon EFS (used for FileStore) does not have similar capabilities. In this case, recovering a database to most recent state but FileStore to an older state may cause inconsistencies between the two.

Based on the backup strategy, the following are options for restore.

- When you restore a backup in AWS Backup, a new resource is created based on the backup that you are restoring. Depending on the component that you restore, you can point you SAP BOBI Platform installation to the new resource or copy data to the original resource. For example, you can restore Amazon FSx and Amazon EBS on a different file system or EBS volume. After you have the new resource available, you can either copy a subset of the data or replace your original resource with the new one. See [Restoring a Backup](#) for details.
- When restoring using third-party software, refer to vendor- and application-specific documentation.
- If you are restoring from Amazon S3 using custom scripts, you will have to restore the backup to an EBS volume, and then use either database specific tools or native operating system features to restore your data back to SAP BOBI Platform installation.

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, if using a Windows Server 2008 R2 or later Windows AMI, the AWS NVMe driver is already included as per the [Amazon EBS and NVMe documentation](#). If you are not using the latest AWS Windows AMIs provided by Amazon, see [Installing or Upgrading AWS NVMe Drivers](#).

Aside from operating system maintenance, there is also maintenance you can consider for EC2 instances themselves. This can be driven via AWS Systems Manager Automation documents. Some examples of this 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 on a schedule by using Amazon CloudWatch Events or by using 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 by 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.

Finally, use the [AWS Instance Scheduler](#) Solution to easily configure custom start and stop schedules for their Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Relational Database Service (Amazon RDS) instances.

Cost Optimization

Just as with right sizing, we recommend customers make cost optimization an ongoing 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 the [Cost Optimization Pillar](#) of the AWS Well-Architected Framework and the [SAP on AWS Pricing and Optimization Guide](#).

Automation

Automation using Infrastructure as Code with AWS CloudFormation

We recommend following the principle of Infrastructure as code (IaC) in 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, and thus follow the principle of IaC.

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.

Integration with AWS Big Data Services

The SAP BOBI Platform product can use multiple AWS Big Data services as data sources for reporting purposes. When using SAP BOBI version 4.2, you can connect to the following AWS data sources:

Table 2: AWS Big Data services support for SAP BusinessObjects Business Intelligence 4.2

Use Case	Amazon Product	SAP BOBI 4.2 Supported
Data source	Amazon RDS Oracle	Yes
Data source	Amazon Redshift	Yes
Data source	Amazon EMR Hive (Hive1)	Yes
Data source	Amazon EMR Hive (Hive2)	Yes
Data source	Amazon EMR Hive 5.6 (Hive2)	Yes

See the [SAP Product Availability Matrix \(PAM\)](#) for the complete list of SAP BOBI Platform supported data sources specific to your version.

Support

To get help from SAP, SAP and AWS requires a business support agreement on AWS. [AWS Business Support](#) provides resources and technical support for customers running SAP workloads on AWS. If you have any technical issues around AWS, you can open up a case with SAP or AWS and it will be routed to the appropriate team. AWS also offers [AWS Enterprise Support](#) for customers running mission critical production workloads on AWS.

Document Revisions

Date	Change
January 2023	Updates throughout the guide
October 2019	Initial publication

SAP BusinessObjects BI Platform on AWS: HA/DR Guide for Linux

SAP specialists, Amazon Web Services (AWS)

Last updated: January 2023

The purpose of this guide is to provide an overview of how to configure high availability (HA) and disaster recovery (DR) for SAP BusinessObjects Business Intelligence (BI) Platform on AWS. This guide will explore how features native to AWS in combination with SAP BusinessObjects BI Platform installation and configuration techniques can greatly improve the availability of an SAP deployment. This guide is not an exhaustive list of all possible configuration options, but covers solutions common to typical deployment scenarios.

This guide isn't intended to replace the SAP BusinessObjects BI Platform installation and administration guides, operating system documentation, or RDBMS documentation.

The procedures and examples in this guide are based on the following:

- A typical, large-scale deployment on AWS that includes two Availability Zones and three subnets in each Availability Zone. You can change this configuration to support your own requirements for SAP BusinessObjects BI Platform servers and tiers.
- An internal Application Load Balancer in front of the web servers, but you can use another internal or internet-facing load balancer.
- Amazon Relational Database Service (Amazon RDS) for MySQL as an example Central Management Server (CMS) and auditing database for SAP BusinessObjects BI Platform. However, you can use any of the [databases supported by SAP](#). HA configuration instructions for other databases aren't included in this guide; see the database-specific documentation on the SAP website.
- Amazon Elastic File System (Amazon EFS) for input and output filestores.

Note

You must have SAP portal access to view the SAP Notes. For more information, see the [SAP Support website](#).

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](#).

Prerequisite Knowledge

AWS Services

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, see [Getting Started with AWS](#).)

- [Amazon Elastic Compute Cloud \(Amazon EC2\)](#)
- [Amazon Elastic Block Store \(Amazon EBS\)](#)
- [Amazon Virtual Private Cloud \(Amazon VPC\)](#)
- [AWS CloudFormation](#)
- [Amazon EFS](#)
- [Amazon RDS](#)

SAP BusinessObjects BI Platform on AWS

This guide assumes that you're already familiar with implementing and operating SAP solutions on AWS. Please read the SAP notes listed in the following table before continuing. SAP BusinessObjects BI Platform is supported on AWS as described in [SAP Note 2438592](#). All AWS guides for SAP BusinessObjects BI Platform can be found on the [SAP on AWS website](#).

SAP Note	Description
1588667	SAP on AWS: Overview of related SAP notes and web links
1656099	SAP on AWS: Supported products, platforms, and landscapes

SAP Note	Description
2442979	Amazon S3 recommendations for SAP BusinessObjects Business Intelligence Platform
2438592	BI Platform 4.2 Cloud Support

High Availability

HA design for a software application protects single points of failure (SPOFs). A SPOF is a critical component of an application whose failure can cause service outage for users. The server-side architecture of SAP BusinessObjects BI Platform consists of five tiers: web, management, storage, processing, and data. (For details, see the administrator's guide on the [SAP BusinessObjects Business Intelligence Platform](#) website.) SAP BusinessObjects BI Platform 4.x provides HA of platform services using CMS clustering, which could provide customers with the required level of redundancy. CMS servers are in the management tier.

SAP BusinessObjects BI Platform tiers are designed as follows to eliminate SPOFs and to provide redundant installation options for native, highly available components:

- **Management tier:** Includes the CMS servers, event servers, and associated services. SAP BusinessObjects BI Platform cannot function without a CMS server. Multiple CMS servers can run in a cluster on different machines. A cluster consists of two or more CMS servers working together on a common CMS system database. The CMS is a SPOF, so you must create a cluster of CMS servers running in more than one Availability Zone for a highly available design.
- **Storage tier:** Includes input and output file repository servers. Install these servers redundantly so that failure of any single server doesn't cause a service outage. The file system used by these servers to store files such as documents, reports, and universes must be on a shared file system. This file system is a SPOF and therefore must be highly available.
- **Web tier and processing tier:** These tiers perform functions like receiving and processing user requests. These tiers are not SPOFs. However, if a server that provides a specific service isn't available, users cannot use that service. To avoid such situations, install these servers redundantly so that the failure of any single server doesn't cause a service outage.

- **Data tier:** Consists of the CMS system database and the auditing data store. The CMS database is a SPOF. Install a highly available database using vendor-specific database HA technologies. The specific method used depends on the type of CMS database you're using.

HA for SAP BusinessObjects BI Platform on AWS

In this guide, we'll provide an example HA architecture that closely resembles a typical on-premises installation, and we'll also show how AWS features in combination with SAP BusinessObjects BI Platform installation options support an HA solution that extends beyond a single data center. For the AWS Cloud, this means that the application is highly available within an AWS Region and can survive the failure of a single Availability Zone. The HA solution discussed in this guide includes these design features:

- SAP BusinessObjects BI Platform nodes are distributed across multiple EC2 instances within a virtual private cloud (VPC). Although Availability Zone failure is a rare occurrence, to protect your application against such scenarios, install each tier of SAP BusinessObjects BI Platform in multiple Availability Zones.
- Applications are deployed on a web server, following best practices from the [Web Application Deployment Guide for Unix](#) on the SAP Help Portal.
- The CMS database is deployed with a failover node that is in a different Availability Zone from the primary node. This guide uses Amazon RDS for MySQL as an example of the SAP BusinessObjects BI Platform CMS database. Amazon RDS is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. In this guide, we use an Amazon RDS MySQL 5.6 Multi-AZ deployment for the CMS database. See the [AWS documentation](#) for details on how you can configure HA for an Amazon RDS MySQL database in a few steps.
- Input and output filestores are deployed on a shared file system such as an Amazon EFS file system. Amazon EFS provides simple, scalable, elastic [file storage](#) for use with AWS Cloud services and on-premises resources.

Note

Amazon EFS isn't available yet in all AWS Regions; see a [list of available regions](#). If you're planning to use a region that doesn't support Amazon EFS, you can choose a partner

network file system (NFS) solution from AWS Marketplace (such as Cloud Volumes ONTAP or SoftNAS) or you can deploy your own solution.

Planning the Deployment in a Primary Region

Good planning is a key step in ensuring a successful HA deployment for SAP BusinessObjects BI Platform on AWS. Consider these guidelines in your planning:

- Perform the sizing exercise using the [SAP BusinessObjects BI4 Sizing Guide](#). You can determine the resource requirements of each tier based on the sizing exercise.
- Create your architecture document for SAP BusinessObjects BI Platform. Based on sizing decisions, determine the distribution of BI components across EC2 instances and subnets. The level of redundancy in your distributed architecture depends on your HA requirements: your recovery time objective (RTO) and recovery point objective (RPO). For example, if you design your system to be available at full capacity during an Availability Zone failure with zero RTO, you can deploy your system so that servers in a single Availability Zone can process all user requests. If your business can tolerate losing partial capacity temporarily, you can provision lost instances by using Amazon Machine Images (AMIs) and AWS CloudFormation templates in the Availability Zone that's available at the time. There may be other options as well, depending on your cost and recovery time requirements.
- Choose a CMS database with an HA feature. SAP BusinessObjects BI Platform cannot function if the CMS database isn't available. The method of replication between the primary and standby databases depends on your RTO and RPO requirements, and must be consistent with your application recovery times. If you use an Amazon RDS database as the CMS database, AWS manages HA setup and failover, as explained later in this guide.
- Design the Amazon VPC IP address range, CIDR block, and subnet ranges before you start the installation.

Designing Network and Security Groups for the Primary Region

Define the security groups that will be used to control access to instances for administrative functions, application and DB-level communications, and isolation of failed resources.

Note

Security groups are firewall rules that you define at the instance or network interface level to open or close specific ports for network communication. You'll need to come up with your own set of rules and configure these based on your application connectivity, setup, and integration requirements.

Here are some of the key considerations for configuring security groups for SAP BusinessObjects BI Platform:

- Users will connect to the web server with web browsers or CMS servers and by using desktop SAP BusinessObjects BI Platform client tools.
- Web servers will communicate with CMS servers and other BusinessObjects BI Platform servers.
- CMS servers will communicate with the CMS and auditing databases.
- The BusinessObjects BI Platform processing tier servers will communicate with the data sources. Data sources could be SAP or non-SAP systems in your landscape where SAP BusinessObjects BI Platform runs the reports.

See [SAP Note 2276646](#) to find out the ports used by different SAP BusinessObjects BI Platform components for communication. The SAP deployment and networking teams should work closely to understand what network traffic to allow in each tier and to configure the tiers accordingly. The following ideas should help provide some structure and guidance:

- Set up a virtual private gateway and one customer gateway. These provide VPN connectivity between the corporate data center and the VPC.
- Set up route table configurations for all the traffic to and from the corporate data center over the VPN tunnel.
- Define all communications on required protocols and ports by using network access control lists (ACLs).
- Set up security groups on management servers with restricted access from certain on-premises networks or IP addresses.
- Set up security groups with limited inbound and outbound protocols and ports for each instance.

Note

Servers within a particular VPC subnet might need to access resources on the internet for actions such as software updates. You can provide this access by adding an internet gateway to the VPC and using a network address translation (NAT) gateway or NAT instance placed within a public subnet to protect internal resources. Another method is to create network routes to direct the traffic to traverse the VPN tunnel, into the corporate data center, and out through corporate proxy servers. See the blog posts *VPC Subnet Zoning Patterns for SAP on AWS (Part I, II and III)* in the [AWS for SAP](#) blog for guidance on designing VPCs for SAP applications.

This guide uses an example of a typical, large-scale deployment with two Availability Zones to maximize availability and durability, and three subnets per Availability Zone to distribute different SAP BusinessObjects BI Platform tiers. However, based on your sizing requirements, you can also use more than two Availability Zones to install and distribute the SAP BusinessObjects BI Platform nodes. There are many ways to distribute the servers among Availability Zones, EC2 instances, and subnets. In this example, we've designed the architecture for HA/DR, with the specifications listed in the following table. See the [Disaster Recovery](#) section for details on DR design and setup.

- The web tier is installed in subnet 1 in both Availability Zones.
- The management, storage, and processing tiers are installed in subnet 2 in both Availability Zones.
- The data tier (CMS database) is installed in the DB subnet group in subnet 3 in both Availability Zones.

AWS component	Key consideration	Specifications for primary region	Information used for DR region
Region	Consider latency requirements, distance from end users.	us-west-2 US West (Oregon)	us-east-1 US East (N. Virginia)
Availability Zone 1	Choose at least two Availability Zones.	us-west-2a	us-east-1a

AWS component	Key consideration	Specifications for primary region	Information used for DR region
Availability Zone 2	Choose at least two Availability Zones.	us-west-2b	Availability Zone 2 isn't available for DR in this example.
VPC IP range/CIDR block	Range shouldn't overlap with existing internal IP range. The IP range should be sized appropriately to support the number of hosts and planned growth.	192.168.0.0/16	10.0.0.0/16
Web tier subnet	CIDR IP range for Availability Zone 1 subnets should accommodate future growth.	192.168.5.0/24	10.0.5.0/24
Management, storage, and processing tier subnet		192.168.6.0/24	10.0.6.0/24
DB group subnet		192.168.7.0/24	10.0.7.0/24
Web tier subnet	CIDR IP range for Availability Zone 2 subnets should accommodate future growth.	192.168.8.0/24	Availability Zone 2 isn't available for DR in this example.
Management, storage and processing tier subnet		192.168.9.0/24	

AWS component	Key consideration	Specifications for primary region	Information used for DR region
DB group subnet		192.168.10.0/24	

Note

The AWS Regions, Availability Zones, and IP addresses listed in the table will be used throughout this guide as needed. You can change these details for your own implementation, based on your design requirements.

Figure 1 shows the high-level architecture of all application and network components used in this example. This architecture includes an internal Application Load Balancer between the two web servers, but you can use other load balancers in your design. In this case, the load balancer can route requests only from clients that have access to the VPC. Typically, this will allow access from your corporate network and from the VPC itself. If your users access SAP BusinessObjects BI Platform from the internet, you can choose to deploy an internet-facing load balancer. An internet-facing load balancer has a publicly resolvable DNS name, so it can route requests from clients over the internet to the EC2 instances registered with the load balancer. For configuration steps, see [How do I connect a public-facing load balancer to EC2 instances that have private IP addresses?](#) in the AWS Knowledge Center.

Note

The AWS Regions and Availability Zones shown in the following diagrams are just examples. This architecture can be used in any AWS Region.

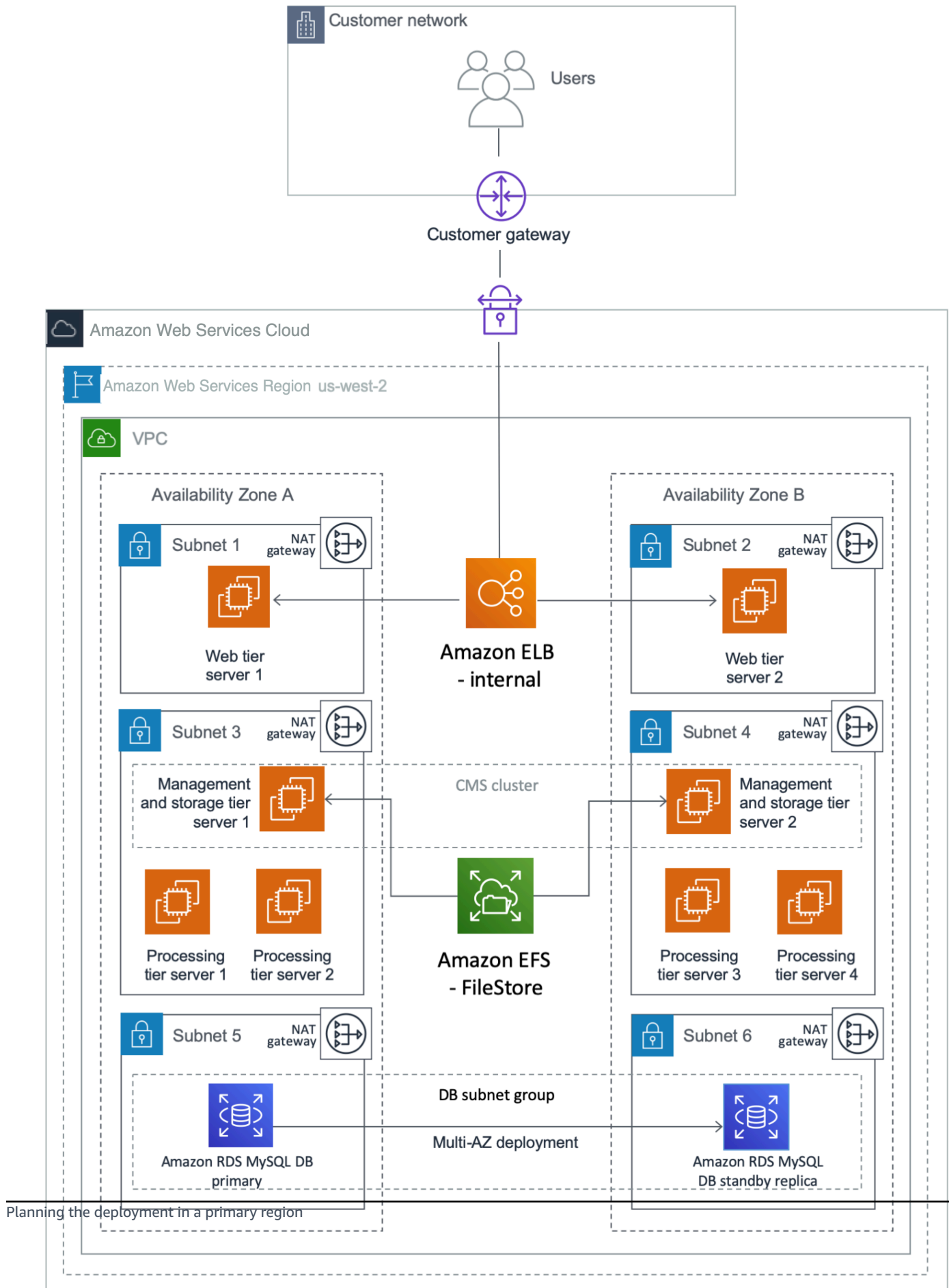


Figure 1: HA architecture for SAP BusinessObjects BI Platform

This example puts the web tier in a dedicated subnet. End users will access the web tier servers where the web applications are deployed. You can use security groups to restrict connectivity between the subnets only to the required ports. For example, you can restrict connectivity from the application subnet to the database subnet to allow access only to the database listener port. The Multi-AZ architecture provides both load balancing and HA.

The Application Load Balancer will distribute the user load across the two web servers in two Availability Zones. The web servers forward the user requests to the CMS servers. Two EC2 instances (one in each Availability Zone) host the management and storage tier. Similarly, there are four instances (two in each Availability Zone) that host the processing tier. You can scale servers in or out further in any tier, according to your requirements. CMS servers installed in both Availability Zones query the CMS database in Availability Zone 1 during normal operations.

In case of an Availability Zone failure, as shown in Figure 2, the health check for the web server in Availability Zone 1 fails, and the Application Load Balancer forwards requests only to the web server in Availability Zone 2. The management and processing tiers in Availability Zone 2 process the user requests. Amazon RDS MySQL switches the standby replica in Availability Zone 2 to primary, and the CMS server connects to the new primary CMS database in Availability Zone 2. If the servers in Availability Zone 2 alone cannot process the user load, you can choose to redeploy additional servers using AMIs and AWS CloudFormation templates.

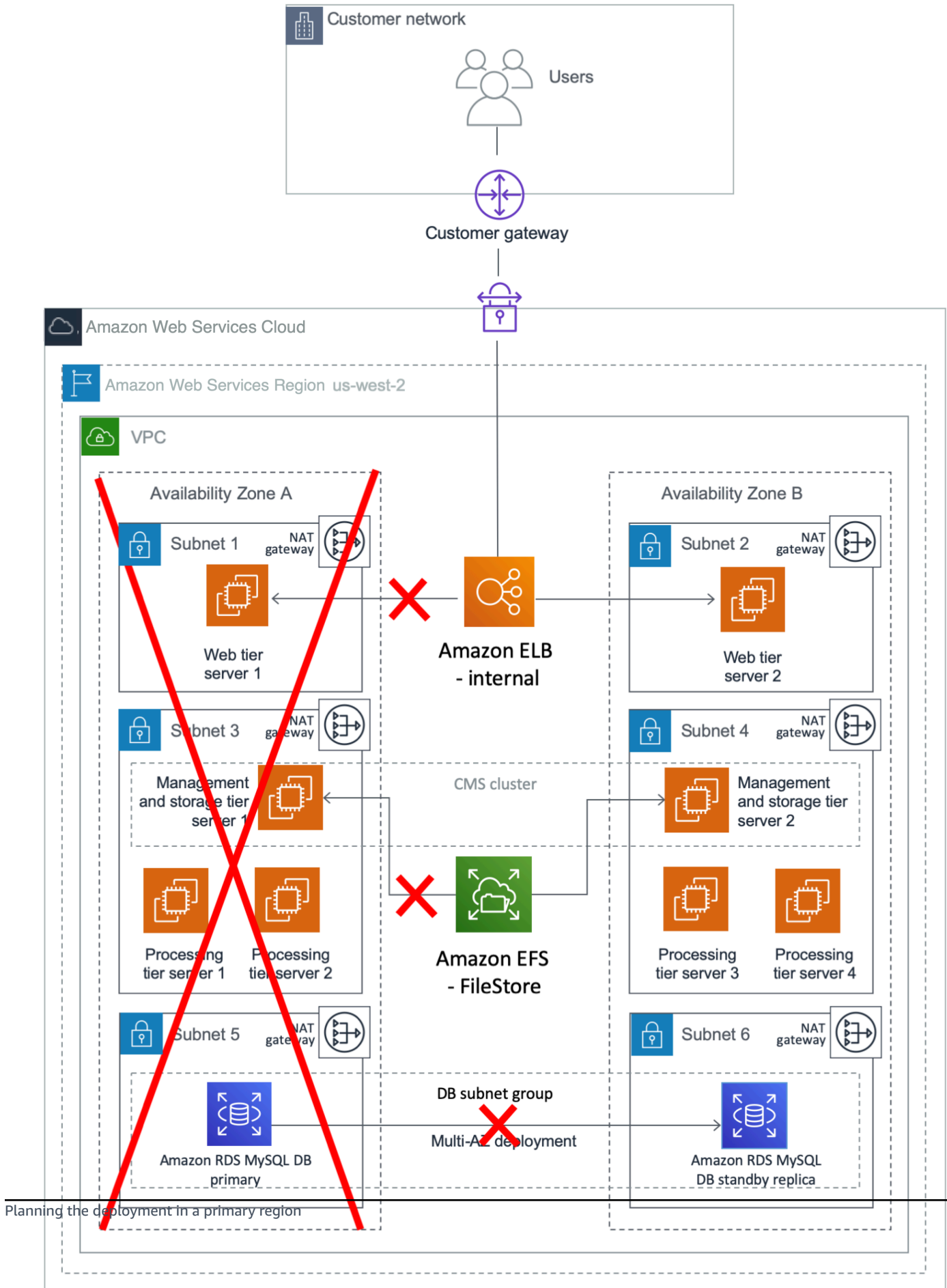


Figure 2: What happens during a failure in Availability Zone 1

Installing SAP BusinessObjects BI Platform for HA

Creating a VPC

1. Create a VPC in one of the available AWS Regions by following the instructions in the [Amazon Virtual Private Cloud documentation](#).

Note

You can also use the [Amazon Virtual Private Cloud Quick Start](#) to provision your VPC.

2. Specify a contiguous IP address range in CIDR block format (e.g., 192.168.0.0/16). You must choose a range that doesn't overlap with an already existing range used internally on your corporate network.
3. Create six new subnets, associate each with the new VPC, and split them across two different Availability Zones.
4. To continue with the installation process, you'll need access to the instances deployed into the VPC. You can use a [VPN connection](#) or [AWS Direct Connect](#) to establish a dedicated network connection from your premises to AWS. In both cases, be sure to create appropriate route tables, associate them with the new subnets, and adjust the network ACLs with rules to meet internal security requirements.

Note

The procedure in this section is based on the large-scale distributed deployment example. You can revise the procedure to support your architecture. For example, you can use additional subnets combined with network ACLs to further isolate or restrict access to different tiers of the SAP BusinessObjects BI Platform environment, and you can create multiple public and private subnets for isolation. Any subnet that is associated with an internet gateway will become a public subnet.

Building the Infrastructure

After you create the VPC, deploy supporting infrastructure instances that will provide key services used by the SAP environment from within the VPC. These services might include:

- Active Directory services (see [Quick Start](#))
- DNS
- Network address translation (NAT) services
- Remote Desktop gateways (see [Quick Start](#))
- Bastion hosts (see [Quick Start](#))

When supporting infrastructure services are in place, you can continue with the next section to start deploying SAP components into the VPC.

Preparing and Installing the CMS Database

This guide uses Amazon RDS MySQL for the CMS database. You can create a separate database for the auditing database if it's required. HA isn't a requirement for the auditing database, so this guide doesn't cover the steps for provisioning it.

1. Create a DB subnet group for an RDS instance by following the instructions in the [Amazon RDS documentation](#). Use two of the subnets you created earlier to create a DB subnet group.
2. In the [Amazon RDS console](#), launch an Amazon RDS MySQL DB instance by following the instructions in the [Amazon RDS documentation](#).
 - Choose a supported DB version based on the [Product Availability Matrix](#) on the SAP website, and select the instance type and storage based on your sizing output.
 - On the **Specify DB details** page, in the **Instance specifications** section, choose **Create replica in different zone**.
 - The **Choose use case** page asks if you are planning to use the DB instance you are creating for production. If you choose **Production - MySQL**, the Multi-AZ failover option is pre-selected.
 - On the **Configure advanced settings** page, provide information about the infrastructure you already provisioned, such as settings for the VPC, DB subnet group, and security group. In addition, you can provide custom options for encryption, backup retention period, maintenance window, and so on. You will also create a user to administer this database.

- For the database name, you can provide the name you want to use for the CMS database. You can also change the database port from the default value of 3306 to your choice of port.
- Choose **Create database**, and then wait for the DB instance status to change to **available** in the Amazon RDS console.
- Choose the **Instances** view and note the **Endpoint** name. In case of failover to another Availability Zone, this endpoint enables an application to reconnect to a new primary database instance without having to change anything.
- (Optional) Create a CNAME in Amazon Route 53 for the database cluster endpoint. Use this CNAME during the installation of SAP BusinessObjects BI Platform nodes.

If you're planning a different database than Amazon RDS MySQL, use the database failover solution appropriate for your use case. Some solutions provide automatic failover on AWS in the event of primary database failure, including the failover of the virtual IP address used for database access. This method provides seamless failover for your applications.

Preparing the OS

Follow these steps to build and configure EC2 instances before installing SAP BusinessObjects BI Platform:

1. Launch a new Amazon Elastic Compute Cloud Linux instance in Availability Zone 1. Prepare the server for the installation of SAP BusinessObjects BI Platform by following the instructions in the [SAP BusinessObjects BI Platform](#) installation guide.
2. Create an EBS volume and attach it to the EC2 Linux instance. For instructions, see the [Amazon EBS user guide](#).
3. Create and mount a file system by following the instructions in the [Amazon EBS user guide](#). Instructions include enabling automatic mounting each time you reboot the Linux instance.
4. Optionally, you can create an AMI of the instance, as described in the [AWS Toolkit for Visual Studio user guide](#), and launch all other SAP BusinessObjects BI Platform nodes by using this AMI.
5. Launch other EC2 instances according to your design, using one of these methods:
 - Manually perform steps 1-3 for all EC2 instances required by your architecture.

- Create an AWS CloudFormation template, using the AMI you created in step 4 as a resource. Use this template to launch all additional instances in their respective Availability Zones. This option saves time and helps prevent manual errors during instance provisioning.

Installing CMS Nodes

1. Log in to each CMS server in the SAP BusinessObjects BI Platform node.
2. Launch the installation as described in the [SAP BusinessObjects BI Platform installation guide](#) and choose:

Custom / Expand > Expand an existing SAP BusinessObjects BI platform deployment > Instances > Servers > Platform Services
3. For the first CMS server installation, choose **Start a new SAP BusinessObjects BI platform deployment**.
4. For all additional CMS server installations, choose **Expand an existing SAP BusinessObjects BI platform deployment**.
5. Enter the details of the first CMS server when prompted. This will cluster the additional CMS servers with the first one.

Note

You can also perform a silent installation by using a response file and the procedure described in the SAP installation guide. This method enables you to quickly rerun the installation again on the same server or on another server, as required. You can further automate the installation by using an AWS CloudFormation template. The template will enable you to provision EC2 instances and install the BusinessObjects BI Platform application in just a few clicks.

Installing the Processing Tier

1. Log in to the processing tier server in the SAP BusinessObjects BI Platform node.
2. Launch the installation as described in the [SAP BusinessObjects BI Platform installation guide](#) and choose:

Custom / Expand > Expand an existing SAP BusinessObjects BI platform deployment > Instances > Servers

3. Choose the servers and other features that you want to install on the processing tier node, and run the installation.
4. Repeat these steps for all your processing tier servers.

Installing the Web Tier

1. Log in to each web tier server in the SAP BusinessObjects BI Platform node.
2. Launch the installation as described in the [SAP BusinessObjects BI Platform installation guide](#) and choose:

Custom / Expand > Expand an existing SAP BusinessObjects BI platform deployment > Instances > WebTier

3. Create a response file for silent installation by choosing **Custom / Expand > Web Tier**. You can install an integrated Tomcat web server by using the SAP installer, or you can use another supported web server and install applications by using the SAP installer.

Post-Installation Configuration for HA

When you have installed the SAP BusinessObjects BI Platform on all servers, you must configure the cluster name, Amazon EFS, and the load balancer to complete the HA configuration.

Configuring the SAP BusinessObjects BI Platform Cluster Name

Set up a cluster name by following the instructions in the [SAP BusinessObjects BI Platform Administrator Guide](#). Use this cluster name in the properties files of all web applications and client tools.

Configuring Amazon EFS for Input and Output Filestores

Note

Amazon EFS isn't available yet in all AWS Regions; see a [list of available regions](#). If you're planning to use a region that doesn't support Amazon EFS, you can choose a partner NFS

solution from AWS Marketplace (such as Cloud Volumes ONTAP or SoftNAS) or you can deploy your own solution.

By default, the [File Repository Server \(FRS\)](#) filestore for SAP BusinessObjects BI Platform is located in the local installation directory. You can set up the filestore on a shared storage system like Amazon EFS, and then mount the Amazon EFS file system on all servers that are configured to run storage tier servers:

1. Create an Amazon EFS file system in the VPC where you installed SAP BusinessObjects BI Platform, and create mount targets in the subnets that run storage tier servers.
2. Configure Amazon EFS for automatic mounting on FRS servers by following the instructions in the [Amazon EFS documentation](#), and run the command **mount -a** to mount the file system.
3. Follow the instructions in the [SAP Community Wiki](#) to change the FRS path to Amazon EFS.

Configuring the Load Balancer for Web Server Access

To distribute the user load evenly across the web tier servers, you can use a load balancer between the web users and the web servers. In this guide, we'll discuss the use of [Elastic Load Balancing \(Elastic Load Balancing\)](#) for this purpose. An Application Load Balancer automatically scales its request handling capacity in response to incoming application traffic. Follow these steps to configure an Application Load Balancer for SAP BusinessObjects BI Platform:

1. Enable Secure Sockets Layer (SSL) communications for SAP BusinessObjects BI Platform by following the instructions in the [administrator guide](#). See also: [Enabling SSL in BI Platform 4.2 SP05](#) on the SAP blog.
2. In the [Amazon EC2 console](#), [create an Application Load Balancer](#) in the VPC where SAP BusinessObjects BI Platform is running. Specify the Availability Zones and subnets of all the web tier servers.
3. [Configure a security group](#) that allows users to connect to the Application Load Balancer on the SSL port.
4. [Create a target group](#) to register web servers as the targets to the load balancer. For **Target type**, choose **ip** and specify the IP address and SSL port of the web servers to register as targets.
5. [Enable sticky sessions](#).

6. (Optional) Create a CNAME in Route 53 for the Application Load Balancer DNS name. Use this CNAME to access SAP BusinessObjects BI Platform.

HA for SAP Data Services

The HA architecture of SAP Data Services is very similar to the HA architecture of SAP BusinessObjects BI Platform. Before you can install SAP Data Services, you must have a working CMS server for SAP BusinessObjects BI Platform. You can install SAP Data Services on existing SAP BusinessObjects BI Platform nodes. If you do not have a SAP BusinessObjects BI Platform installation, SAP BusinessObjects Information Platform Services (IPS) can provide the basic CMS functions required by SAP Data Services. For either option, you can follow the steps in the previous sections to install the CMS server (by clustering two IPS CMS servers together), the CMS database, and a filestore with HA. You must use two Availability Zones for the CMS servers, a highly available shared file system like Amazon EFS, and a CMS database with a Read Replica in a separate Availability Zone. Here are the high-level steps:

1. Create databases for the SAP Data Services repository in Microsoft SQL Server (MS SQL). You can use your existing SAP BusinessObjects BI Platform or SAP BusinessObjects IPS CMS database engine to create a separate database for the repository. Alternatively, you can create a separate MS SQL instance for the repository.
2. Install SAP BusinessObjects BI Platform or SAP BusinessObjects IPS with a minimum of two CMS servers in two separate Availability Zones. See the [SAP Data Services installation guide](#) for IPS installation instructions.
3. Configure the filestore shared file system as described in the [previous section](#).
4. Install SAP Data Services on all SAP BusinessObjects BI Platform servers or IPS servers.
5. Create an SAP Data Services central repository in the database created in step 1 by following the steps in the section *Repository management* in the [SAP Data Services administrator guide](#).
6. Configure HA for SAP Data Services batch jobs by following the instructions in [SAP Note 1938068](#).

HA Testing

[SAP Note 1229417](#) provides four tests you must perform to ensure that the CMSs are clustered correctly and that cluster functionality is working. Additionally, when performing the tests, you must also ensure that your deployment can tolerate Availability Zone failure. You can confirm this

by stopping all the servers in one Availability Zone and making sure that all the services are still available. To test the CMS database, you can perform the failover manually. In the [Amazon RDS console](#), choose **Instances**, **Instance actions**, **Failover**. After all servers in one Availability Zone are shut down and database failover is complete, test the availability of the system and all the services. Repeat the same test for the second Availability Zone.

Disaster Recovery

The DR approach you take for SAP BusinessObjects BI Platform, as for any other enterprise application, depends on your RTO and RPO requirements. As discussed in [SAP Note 2056228](#), there are two options for building a DR site for SAP BusinessObjects BI Platform:

- Fully or selectively using SAP Lifecycle Manager (LCM) or Data Federation Services to promote or distribute the content from the primary system.
- Periodically copying over the CMS database and FRS contents, and using that to start a secondary system when required.

DR for SAP BusinessObjects BI Platform on AWS

DR for SAP BusinessObjects BI Platform on the AWS Cloud refers to a scenario in which the primary AWS Region where SAP BusinessObjects BI Platform application is running is unavailable. The goal of setting up a DR site is to be able to recover the application within your acceptable RTO and RPO.

There are no restrictions in AWS for using LCM or Data Federation Services for your DR environment. Note that using LCM requires either consuming resources on the source system or provisioning an additional system to run promotion management jobs. This option might also result in a higher RPO depending on the frequency of promotion management jobs. See [Promotion Management Architecture](#) on the SAP Community Wiki for the high-level architecture for this option.

In this guide, we'll discuss the second option for handling DR, which is to copy the CMS database and FRS contents. Using variants of this option, you can build the complete primary system within your recovery time limits. This option doesn't require resources from the primary system except for the backup copy of the database and file system.

Planning the Deployment in the DR Region

Planning the deployment of a DR system is critical. Consider these items in your planning:

1. Choose a region for your DR site. Make sure that this DR region supports the services and resources that you are using for your SAP BusinessObjects BI Platform. Each AWS Region is completely isolated from other AWS Regions to provide the best possible fault tolerance and stability. You may choose a DR region that is geographically closer to your users. For the examples in this guide, we'll assume US West (Oregon) as the primary region and US East (N. Virginia) as the DR region.
2. Determine the RTO and RPO that are acceptable to your business. This decision determines the methodologies you choose for copying the CMS database and FRS content from the primary site to the DR site. See [Affordable Enterprise-Grade Disaster Recovery Using AWS](#) on the AWS website for options based on your RTO and RPO requirements. In this guide, because we're using Amazon RDS MySQL for the CMS database, we have two options for copying the database:
 - Perform periodic snapshots of the Amazon RDS MySQL database and copy these snapshots to the DR region (results in lower RTO and RPO, and lower cost).
 - Create an Amazon RDS MySQL Read Replica in the DR region, and promote this Read Replica to master in the event of a disaster or DR drill (results in higher RTO and RPO, and higher cost).

For details, see the section [Installing and Configuring the CMS Database for the DR Region](#) later in this guide.
3. Determine the capacity of the DR site. You might want the DR site to have the same capacity and resources as the primary site, or you might decide to run it with fewer resources.
4. For the examples in this guide, we're using a Route 53 CNAME with an Application Load Balancer for user connectivity. You can change the value of the CNAME to point to the load balancer of the DR region in the event of a disaster or DR drill. This will direct users to the DR environment. You can also use other methods to redirect users if you aren't using Route 53.
5. Determine the failover process. Define the disaster situations beforehand and put a mechanism in place to direct users to the DR site if a disaster occurs. Define test cases to simulate these scenarios, and perform a DR drill.

Designing Network and Security Groups for the DR Region

Network and security groups in the DR region will follow the same design as in the primary region, as discussed in the section [Designing Network and Security Groups for the Primary Region](#) earlier in this guide. The table in that section lists the VPC and subnet CIDRs that the DR region uses for

the examples in this guide. Figure 3 shows the DR architecture used in this guide. This guide does not discuss the use of a secondary Availability Zone for HA in the DR region, but based on your requirements, you can use one to install additional servers.

Multiple Region Multi-VPC Connectivity

See [Multiple Region Multi-VPC Connectivity](#) on the AWS website to find options for connecting multiple VPCs in different AWS Regions. If you aren't using an Amazon RDS database, you can use the database replication method supported by your database vendor to build a DR database and continuously copy the data. SAP HANA System Replication is one such example. AWS provides options to establish a connection between the primary and DR VPCs in two separate AWS Regions securely and reliably. [Multiple Region Multi-VPC Connectivity](#) discusses two ways to connect the VPCs:

- **Routing over AWS networks:** This option enables you to use an AWS backbone network for connectivity. AWS provides a high bandwidth, global network infrastructure to support your regional networking needs. AWS Regions are connected to multiple Internet Service Providers (ISPs) and to a private global network backbone, which provides lower cost and more consistent cross-region network latency when compared with the public internet.
- **Routing over non-AWS networks:** This option enables you to route DR network traffic through your corporate data center. Customers can use an internet-based VPN or their own corporate network backbone.

Each option provides different benefits for bandwidth reliability, security, HA, and use of existing infrastructure, and might have limitations as well. You can choose the option that meets your requirements. You can also use AWS Direct Connect gateways to connect your AWS Direct Connect connection over a private virtual interface to one or more VPCs located in the same or different regions in your account. See [AWS Direct Connect Gateways](#) in the AWS documentation for the rules and limitations of using this approach.

Note

The AWS Regions and Availability Zones shown in the following diagrams are just examples. This architecture can be used in any AWS Region.

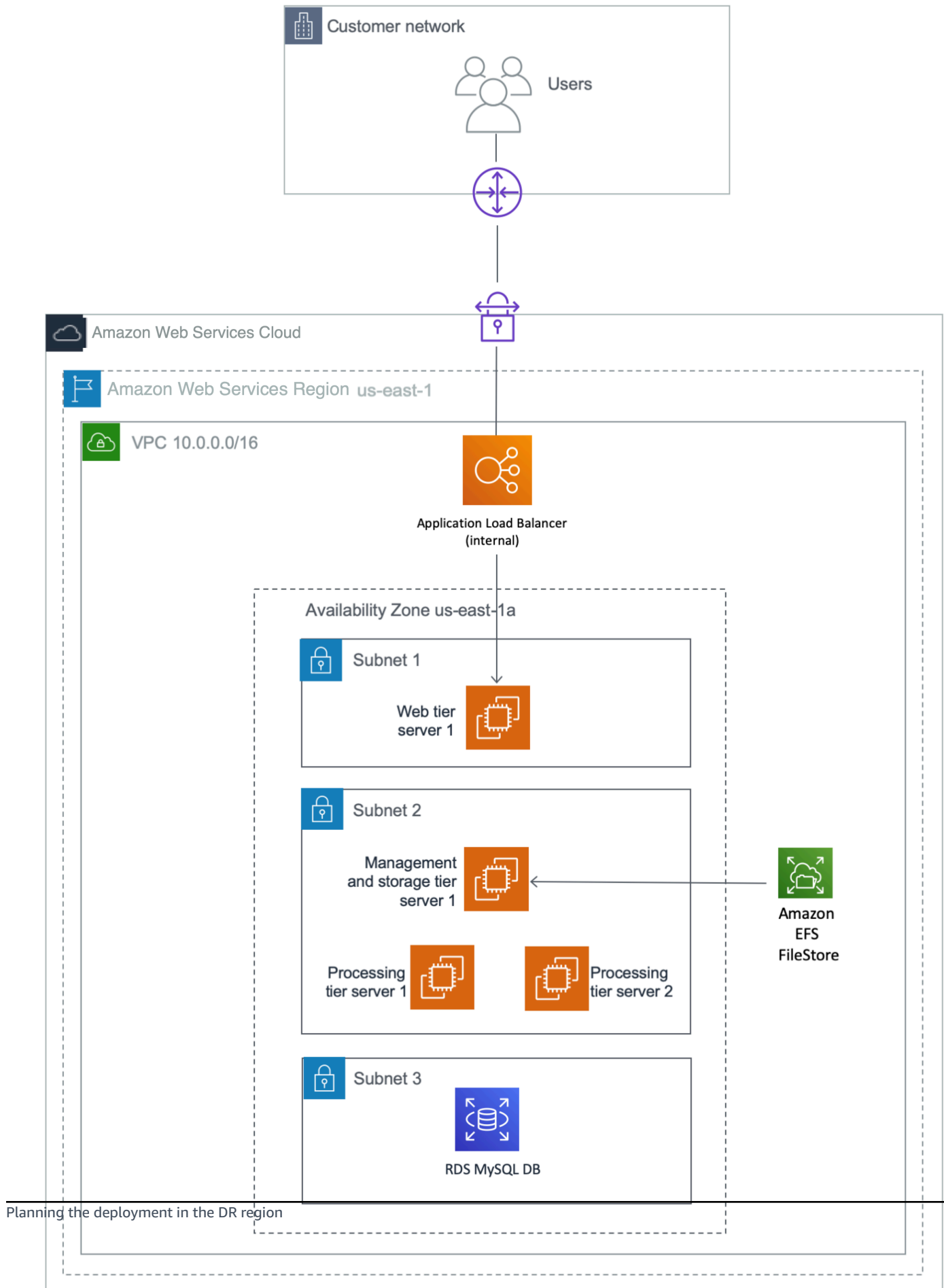


Figure 3: DR architecture for SAP BusinessObjects BI Platform

Figure 4 shows the user connectivity using Route 53 as an example. You can also use your own DNS. During normal operations, the value of the CNAME record for SAP BusinessObjects BI Platform points to the Application Load Balancer of the primary region. The load balancer distributes the load between the available web servers that deploy the SAP BusinessObjects BI Platform web applications.

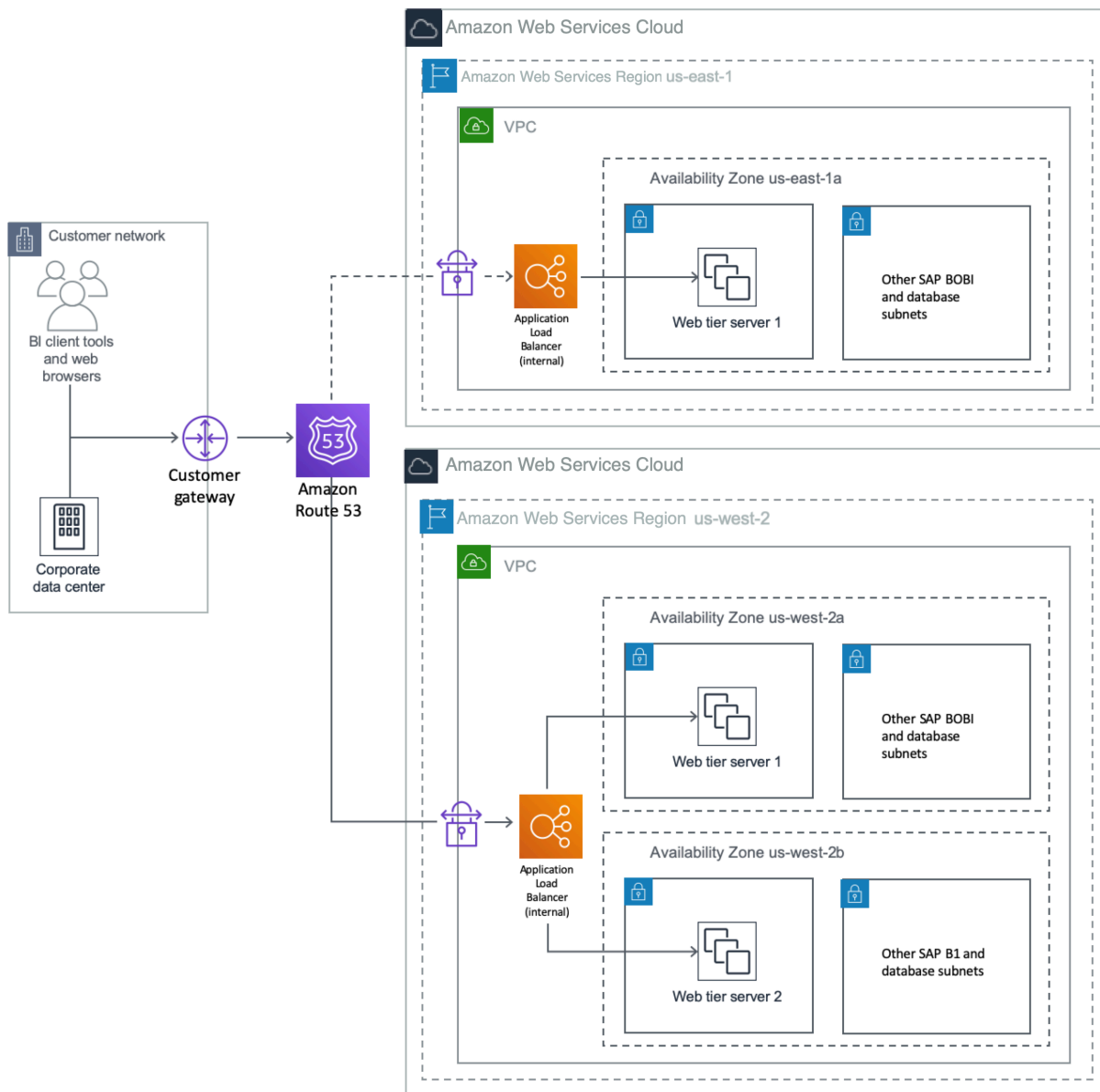


Figure 4: User connectivity with Route 53

In the event of DR or when performing a DR drill, you can either manually switch the DNS to point to the DR region or you can use automatic failover by using Route 53 health checks. For details, see [Configuring DNS Failover](#) in the Route 53 documentation. Redirect all the users to the DR region as shown in Figure 5.

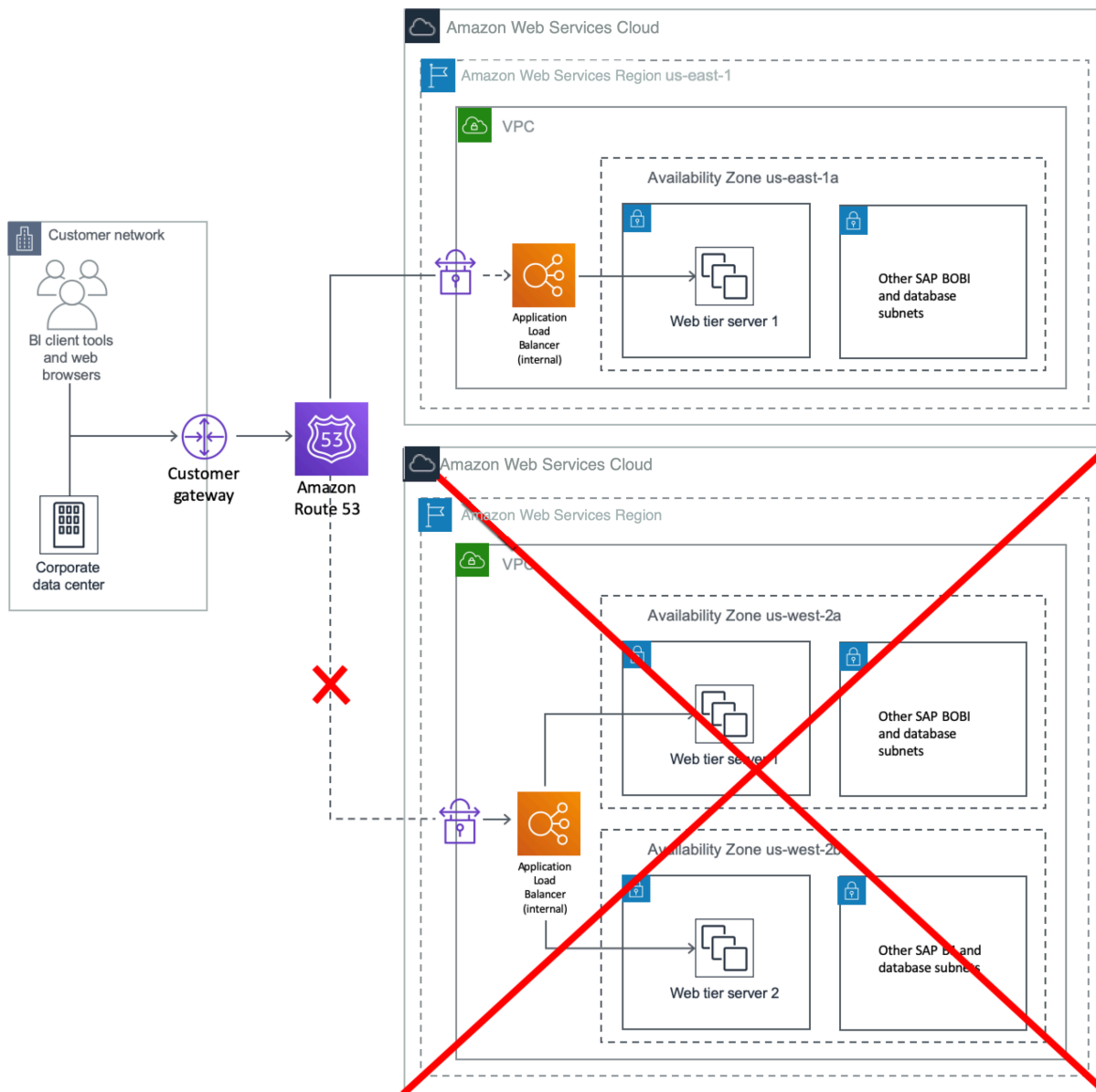


Figure 5: Redirecting users to a DR region

Installing SAP BusinessObjects BI Platform in the DR Region

When you have selected the DR region, follow the steps in the [Preparing the OS](#) section earlier in this guide to build SAP BusinessObjects BI Platform EC2 instances in that region.

Installing and Configuring the CMS Database for the DR Region

The CMS database in the DR region must be a copy of the CMS database in the primary region. The method for copying this database depends on the database type and your RTO and RPO requirements. This guide describes two options for an Amazon RDS MySQL database.

- Option 1: Create a cross-region Read Replica of the primary Amazon RDS MySQL database.

Amazon RDS uses the MySQL DB engine's built-in replication functionality to create a special type of DB instance called a Read Replica from a source DB instance. Updates made to the source DB instance are asynchronously copied to the Read Replica. Create a Read Replica of the primary CMS database by following the instructions in the [Amazon RDS documentation](#). The documentation also provides instructions for promoting a Read Replica to a primary DB instance. This option will provide a lower RPO and RTO than Option 2.

- Option 2: Copy primary database snapshots to the DR region and use them to build a DR CMS database.

Create a snapshot of the Amazon RDS MySQL DB and copy the snapshot to the DR region. Use this snapshot to create a new instance of the Amazon RDS MySQL DB, and use this instance to configure the SAP BusinessObjects BI Platform DR database. See the [Amazon RDS documentation](#) for instructions, features, and limitations of this option. For a continuous copy of the database content, you can create and schedule scripts to perform the snapshot and copy them to the DR region. You can also use open-source tools to deploy such solutions, but make sure to test any tools thoroughly before using them for your DR deployment. When your testing is complete and the primary region is active, you can delete the RDS instance. Create a new instance by using the latest snapshot when the DR system is required again.

Installing a CMS Node

The steps to install the SAP BusinessObjects BI Platform application in the DR region are described in [SAP Note 2603573](#). This procedure uses SQL Anywhere to temporarily create Server Intelligence Agents (SIAs) on all instances with a local CMS database. In the next section, we'll reconfigure all nodes to form a single SAP BusinessObjects BI Platform cluster.

1. Log in to all CMS servers for your SAP BusinessObjects BI Platform DR nodes, based on your design.
2. Install the CMS servers by following the instructions in the SAP Note and using the same SIA names as in the production cluster.

Starting SAP BusinessObjects BI Platform in the DR Region

Create the SIA nodes for the CMS servers by following the procedure described in [SAP Note 2603573](#). When prompted for the CMS database details, use the connection information and credentials of the database created in the previous section. The CMS servers should now start correctly. You can now install the remaining tiers, such as the processing tier or web tier, as required by your design. The installation instructions for the primary region also apply to the DR region.

Creating an Amazon EFS Filestore for the DR Region

The filestore for the DR region in the Amazon EFS file system must be a copy of the filestore in the primary region. You can use AWS DataSync for copying this data. See [Transfer Files to Amazon EFS Using AWS DataSync](#) for detailed steps.

1. Set up correct permissions for the Amazon EFS file system for the filestore.
2. Restart the input and output file servers in the DR region. You should now be able to verify that the SAP BusinessObjects BI Platform system in the DR region has all the data of the primary region, as of the time the database and filestore were last copied.

Performing the Failover

When you complete all the steps described in the previous sections, the SAP BusinessObjects BI Platform system will be up and running in the DR region with primary region data. You can shut down the EC2 instances in the DR region when you don't need them. Monitor the copy of the database and Amazon EFS file system to make sure that they're active according to your configuration. In the event of a disaster or DR drill, follow these steps:

1. Start the CMS database in the DR region using one of these options:
 - If you're using a Read Replica, open the [Amazon RDS console](#) in the DR region, and then choose **Instances**. Select the DB instance that is the Read Replica of the production

database, and then choose **Instance Actions, Promote Read Replica** to start the CMS database in the DR region.

- If you're using snapshots, open the [Amazon RDS console](#) in the DR region, and then choose **Snapshots**. Select the latest snapshot and restore that snapshot to create a CMS database in the DR region.
- 2. If you're using snapshots, update the data source settings for SAP BusinessObjects BI Platform in the DR region by following the instructions in the section *To select a new or existing CMS database on UNIX* in the [SAP BusinessObjects BI Platform Administration Guide](#). Use the RDS instance created with the latest CMS database snapshot in the primary region.
- 3. Copy the contents of the S3 bucket that has the latest Amazon EFS content (from the primary region) to the Amazon EFS file system for the input and output filestore, and set up correct permissions.
- 4. Start SAP BusinessObjects BI Platform in the DR region and perform validation.
- 5. Redirect your users by changing the value of the CNAME used for SAP BusinessObjects BI Platform from the Application Load Balancer for the primary region to the load balancer for the DR region.

DR for SAP Data Services

As described earlier in the [HA for SAP Data Services](#) section, you can install SAP Data Services on either an existing SAP BusinessObjects BI Platform node or on SAP BusinessObjects Information Platform Services (IPS). In both cases, the DR approach described previously in this section applies to SAP Data Services as well.

Summary

SAP customers can now deploy SAP BusinessObjects BI Platform solution and landscapes on the scalable, on-demand Amazon EC2 platform in a highly available manner without having to invest in costly capital expenditures for the underlying infrastructure. By combining the flexibility of the AWS platform with SAP installation techniques, SAP customers can greatly improve the availability of their deployments. For further information, including case studies of customers who have deployed SAP systems on AWS, see [SAP Case Studies](#) on the AWS website.

Additional Tips

Tagging AWS Resources

Adding tags to AWS objects will make it much easier to manage the SAP HA environment, and can also help you search for resources quickly. You can use Amazon EC2 API calls in conjunction with a special tag filter. For more information about tagging resources, see the [AWS documentation](#).

Third-Party Software Components

Additional third-party components might be integral to running business processes within an SAP environment. After you determine your requirements, consider leveraging some of the concepts discussed in this guide, such as:

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

SAP BusinessObjects BI Platform on AWS: HA/DR Guide for Windows

SAP specialists, Amazon Web Services (AWS)

Last updated: January 2023

The purpose of this guide is to provide an overview of how to configure high availability (HA) and disaster recovery (DR) for SAP BusinessObjects Business Intelligence (BI) Platform on AWS. This guide will explore how features native to AWS in combination with SAP BusinessObjects BI Platform installation and configuration techniques can greatly improve the availability of an SAP deployment. This guide is not an exhaustive list of all possible configuration options, but covers solutions common to typical deployment scenarios.

This guide isn't intended to replace the SAP BusinessObjects BI Platform installation and administration guides, operating system documentation, or RDBMS documentation.

The procedures and examples in this guide are based on the following:

- A typical, large-scale deployment on AWS that includes two Availability Zones and three subnets in each Availability Zone. You can change this configuration to support your own requirements for SAP BusinessObjects BI Platform servers and tiers.
- An internal Application Load Balancer in front of the web servers, but you can use another internal or internet-facing load balancer.
- Amazon Relational Database Service (Amazon RDS) for MySQL as an example Central Management Server (CMS) and auditing database for SAP BusinessObjects BI Platform. However, you can use any of the [databases supported by SAP](#). HA configuration instructions for other databases aren't included in this guide; see the database-specific documentation on the SAP website.
- Amazon FSx for Windows File Server for input and output filestores.

Note

You must have SAP portal access to view the SAP Notes. For more information, see the [SAP Support website](#).

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](#).

Prerequisite Knowledge

AWS Services

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, see [Getting Started with AWS](#).)

- [Amazon Elastic Compute Cloud \(Amazon EC2\)](#)
- [Amazon Elastic Block Store \(Amazon EBS\)](#)
- [Amazon Virtual Private Cloud \(Amazon VPC\)](#)
- [AWS CloudFormation](#)
- [Amazon FSx](#)
- [Amazon RDS](#)

SAP BusinessObjects BI Platform on AWS

This guide assumes that you're already familiar with implementing and operating SAP solutions on AWS. Please read the SAP notes listed in the following table before continuing. SAP BusinessObjects BI Platform is supported on AWS as described in [SAP Note 2438592](#). All AWS guides for SAP BusinessObjects BI Platform can be found on the [SAP on AWS website](#).

SAP Note	Description
1588667	SAP on AWS: Overview of related SAP notes and web links
1656099	SAP on AWS: Supported products, platforms, and landscapes

SAP Note	Description
2442979	Amazon S3 recommendations for SAP BusinessObjects Business Intelligence Platform
2438592	BI Platform 4.2 Cloud Support

High Availability

HA design for a software application protects single points of failure (SPOFs). A SPOF is a critical component of an application whose failure can cause service outage for users. The server-side architecture of SAP BusinessObjects BI Platform consists of five tiers: web, management, storage, processing, and data. (For details, see the administrator's guide on the [SAP BusinessObjects Business Intelligence Platform](#) website.) SAP BusinessObjects BI Platform 4.x provides HA of platform services using CMS clustering, which could provide customers with the required level of redundancy. CMS servers are in the management tier.

SAP BusinessObjects BI Platform tiers are designed as follows to eliminate SPOFs and to provide redundant installation options for native, highly available components:

- **Management tier:** Includes the CMS servers, event servers, and associated services. SAP BusinessObjects BI Platform cannot function without a CMS server. Multiple CMS servers can run in a cluster on different machines. A cluster consists of two or more CMS servers working together on a common CMS system database. The CMS is a SPOF, so you must create a cluster of CMS servers running in more than one Availability Zone for a highly available design.
- **Storage tier:** Includes input and output file repository servers. Install these servers redundantly so that failure of any single server doesn't cause a service outage. The file system used by these servers to store files such as documents, reports, and universes must be on a shared file system. This file system is a SPOF and therefore must be highly available.
- **Web tier and processing tier:** These tiers perform functions like receiving and processing user requests. These tiers are not SPOFs. However, if a server that provides a specific service isn't available, users cannot use that service. To avoid such situations, install these servers redundantly so that the failure of any single server doesn't cause a service outage.

- **Data tier:** Consists of the CMS system database and the auditing data store. The CMS database is a SPOF. Install a highly available database using vendor-specific database HA technologies. The specific method used depends on the type of CMS database you're using.

HA for SAP BusinessObjects BI Platform on AWS

In this guide, we'll provide an example HA architecture that closely resembles a typical on-premises installation, and we'll also show how AWS features in combination with SAP BusinessObjects BI Platform installation options support an HA solution that extends beyond a single data center. For the AWS Cloud, this means that the application is highly available within an AWS Region and can survive the failure of a single Availability Zone. The HA solution discussed in this guide includes these design features:

- SAP BusinessObjects BI Platform nodes are distributed across multiple EC2 instances within a virtual private cloud (VPC). Although Availability Zone failure is a rare occurrence, to protect your application against such scenarios, install each tier of SAP BusinessObjects BI Platform in multiple Availability Zones.
- Applications are deployed on a web server, following best practices from the [Web Application Deployment Guide for Unix](#) on the SAP Help Portal.
- The CMS database is deployed with a failover node that is in a different Availability Zone from the primary node. This guide uses Amazon RDS for MySQL as an example of the SAP BusinessObjects BI Platform CMS database. Amazon RDS is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. In this guide, we use an Amazon RDS MySQL 5.6 Multi-AZ deployment for the CMS database. See the [AWS documentation](#) for details on how you can configure HA for an Amazon RDS MySQL database in a few steps.
- Input and output filestores are deployed on a shared file system such as Amazon FSx. FSx for Windows File Server a fully managed native Microsoft Windows file system so you can easily move your Windows-based applications that require file storage to AWS.

Note

Amazon FSx isn't available yet in all AWS Regions; see a [list of available regions](#). If you're planning to use a region that doesn't support Amazon FSx, you can choose a partner

network file system (NFS) solution from AWS Marketplace (such as Cloud Volumes ONTAP or SoftNAS) or you can deploy your own solution.

Planning the Deployment in a Primary Region

Good planning is a key step in ensuring a successful HA deployment for SAP BusinessObjects BI Platform on AWS. Consider these guidelines in your planning:

- Perform the sizing exercise using the [SAP BusinessObjects BI4 Sizing Guide](#). You can determine the resource requirements of each tier based on the sizing exercise.
- Create your architecture document for SAP BusinessObjects BI Platform. Based on sizing decisions, determine the distribution of BI components across EC2 instances and subnets. The level of redundancy in your distributed architecture depends on your HA requirements: your recovery time objective (RTO) and recovery point objective (RPO). For example, if you design your system to be available at full capacity during an Availability Zone failure with zero RTO, you can deploy your system so that servers in a single Availability Zone can process all user requests. If your business can tolerate losing partial capacity temporarily, you can provision lost instances by using Amazon Machine Images (AMIs) and AWS CloudFormation templates in the Availability Zone that's available at the time. There may be other options as well, depending on your cost and recovery time requirements.
- Choose a CMS database with an HA feature. SAP BusinessObjects BI Platform cannot function if the CMS database isn't available. The method of replication between the primary and standby databases depends on your RTO and RPO requirements, and must be consistent with your application recovery times. If you use an Amazon RDS database as the CMS database, AWS manages HA setup and failover, as explained later in this guide.
- Design the Amazon VPC IP address range, CIDR block, and subnet ranges before you start the installation.

Designing Network and Security Groups for the Primary Region

Define the security groups that will be used to control access to instances for administrative functions, application and DB-level communications, and isolation of failed resources.

Note

Security groups are firewall rules that you define at the instance or network interface level to open or close specific ports for network communication. You'll need to come up with your own set of rules and configure these based on your application connectivity, setup, and integration requirements.

Here are some of the key considerations for configuring security groups for SAP BusinessObjects BI Platform:

- Users will connect to the web server with web browsers or CMS servers and by using desktop SAP BusinessObjects BI Platform client tools.
- Web servers will communicate with CMS servers and other BusinessObjects BI Platform servers.
- CMS servers will communicate with the CMS and auditing databases.
- The BusinessObjects BI Platform processing tier servers will communicate with the data sources. Data sources could be SAP or non-SAP systems in your landscape where SAP BusinessObjects BI Platform runs the reports.

See [SAP Note 2276646](#) to find out the ports used by different SAP BusinessObjects BI Platform components for communication. The SAP deployment and networking teams should work closely to understand what network traffic to allow in each tier and to configure the tiers accordingly. The following ideas should help provide some structure and guidance:

- Set up a virtual private gateway and one customer gateway. These provide VPN connectivity between the corporate data center and the VPC.
- Set up route table configurations for all the traffic to and from the corporate data center over the VPN tunnel.
- Define all communications on required protocols and ports by using network access control lists (ACLs).
- Set up security groups on management servers with restricted access from certain on-premises networks or IP addresses.
- Set up security groups with limited inbound and outbound protocols and ports for each instance.

Note

Servers within a particular VPC subnet might need to access resources on the internet for actions such as software updates. You can provide this access by adding an internet gateway to the VPC and using a network address translation (NAT) gateway or NAT instance placed within a public subnet to protect internal resources. Another method is to create network routes to direct the traffic to traverse the VPN tunnel, into the corporate data center, and out through corporate proxy servers. See the blog posts *VPC Subnet Zoning Patterns for SAP on AWS (Part I, II and III)* in the [AWS for SAP](#) blog for guidance on designing VPCs for SAP applications.

This guide uses an example of a typical, large-scale deployment with two Availability Zones to maximize availability and durability, and three subnets per Availability Zone to distribute different SAP BusinessObjects BI Platform tiers. However, based on your sizing requirements, you can also use more than two Availability Zones to install and distribute the SAP BusinessObjects BI Platform nodes. There are many ways to distribute the servers among Availability Zones, EC2 instances, and subnets. In this example, we've designed the architecture for HA/DR, with the specifications listed in the following table. See the [Disaster Recovery](#) section for details on DR design and setup.

- The web tier is installed in subnet 1 in both Availability Zones.
- The management, storage, and processing tiers are installed in subnet 2 in both Availability Zones.
- The data tier (CMS database) is installed in the DB subnet group in subnet 3 in both Availability Zones.

Example design for HA/DR architecture

AWS component	Key consideration	Specifications for primary region	Information used for DR region
Region	Consider latency requirements, distance from end users.	us-west-2 US West (Oregon)	us-east-1 US East (N. Virginia)

AWS component	Key consideration	Specifications for primary region	Information used for DR region
Availability Zone 1	Choose at least two Availability Zones.	us-west-2a	us-east-1a
Availability Zone 2	Choose at least two Availability Zones.	us-west-2b	Availability Zone 2 isn't available for DR in this example.
VPC IP range/CIDR block	Range shouldn't overlap with existing internal IP range. The IP range should be sized appropriately to support the number of hosts and planned growth.	192.168.0.0/16	10.0.0.0/16
Web tier subnet	CIDR IP range for Availability Zone 1 subnets should accommodate future growth.	192.168.5.0/24	10.0.5.0/24
Management, storage, and processing tier subnet		192.168.6.0/24	10.0.6.0/24
DB group subnet		192.168.7.0/24	10.0.7.0/24
Web tier subnet	CIDR IP range for Availability Zone 2 subnets should accommodate future growth.	192.168.8.0/24	Availability Zone 2 isn't available for DR in this example.

AWS component	Key consideration	Specifications for primary region	Information used for DR region
Management, storage and processing tier subnet		192.168.9.0/24	
DB group subnet		192.168.10.0/24	

Note

The AWS Regions, Availability Zones, and IP addresses listed in the table will be used throughout this guide as needed. You can change these details for your own implementation, based on your design requirements.

Figure 1 shows the high-level architecture of all application and network components used in this example. This architecture includes an internal Application Load Balancer between the two web servers, but you can use other load balancers in your design. In this case, the load balancer can route requests only from clients that have access to the VPC. Typically, this will allow access from your corporate network and from the VPC itself. If your users access SAP BusinessObjects BI Platform from the internet, you can choose to deploy an internet-facing load balancer. An internet-facing load balancer has a publicly resolvable DNS name, so it can route requests from clients over the internet to the EC2 instances registered with the load balancer. For configuration steps, see [How do I connect a public-facing load balancer to EC2 instances that have private IP addresses?](#) in the AWS Knowledge Center.

Note

The AWS Regions and Availability Zones shown in the following diagrams are just examples. This architecture can be used in any AWS Region.

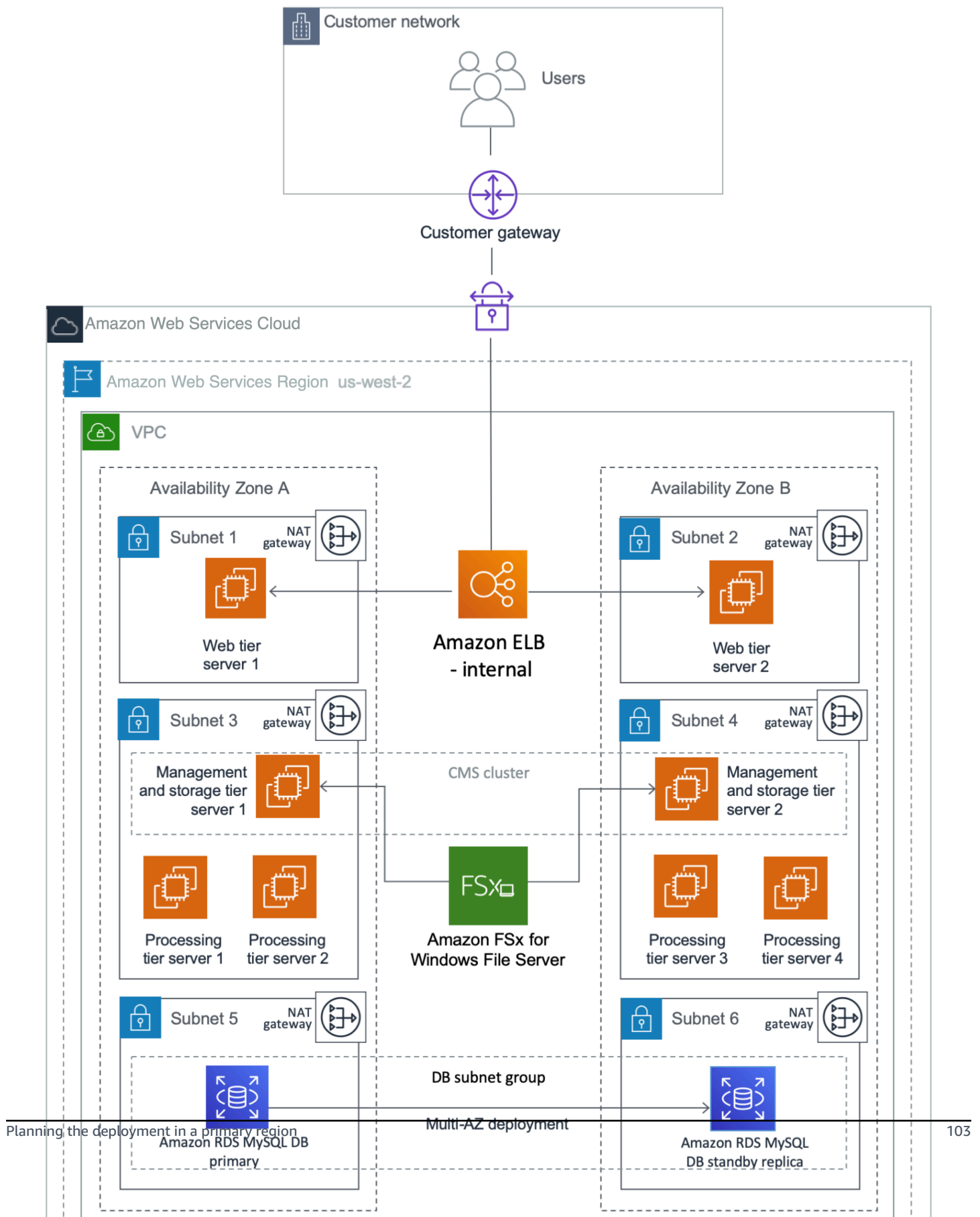


Figure 1: HA architecture for SAP BusinessObjects BI Platform

This example puts the web tier in a dedicated subnet. End users will access the web tier servers where the web applications are deployed. You can use security groups to restrict connectivity between the subnets only to the required ports. For example, you can restrict connectivity from the application subnet to the database subnet to allow access only to the database listener port. The Multi-AZ architecture provides both load balancing and HA.

The Application Load Balancer will distribute the user load across the two web servers in two Availability Zones. The web servers forward the user requests to the CMS servers. Two EC2 instances (one in each Availability Zone) host the management and storage tier. Similarly, there are four instances (two in each Availability Zone) that host the processing tier. You can scale servers in or out further in any tier, according to your requirements. CMS servers installed in both Availability Zones query the CMS database in Availability Zone 1 during normal operations.

In case of an Availability Zone failure, as shown in Figure 2, the health check for the web server in Availability Zone 1 fails, and the Application Load Balancer forwards requests only to the web server in Availability Zone 2. The management and processing tiers in Availability Zone 2 process the user requests. Amazon RDS MySQL switches the standby replica in Availability Zone 2 to primary, and the CMS server connects to the new primary CMS database in Availability Zone 2. If the servers in Availability Zone 2 alone cannot process the user load, you can choose to redeploy additional servers using AMIs and AWS CloudFormation templates.

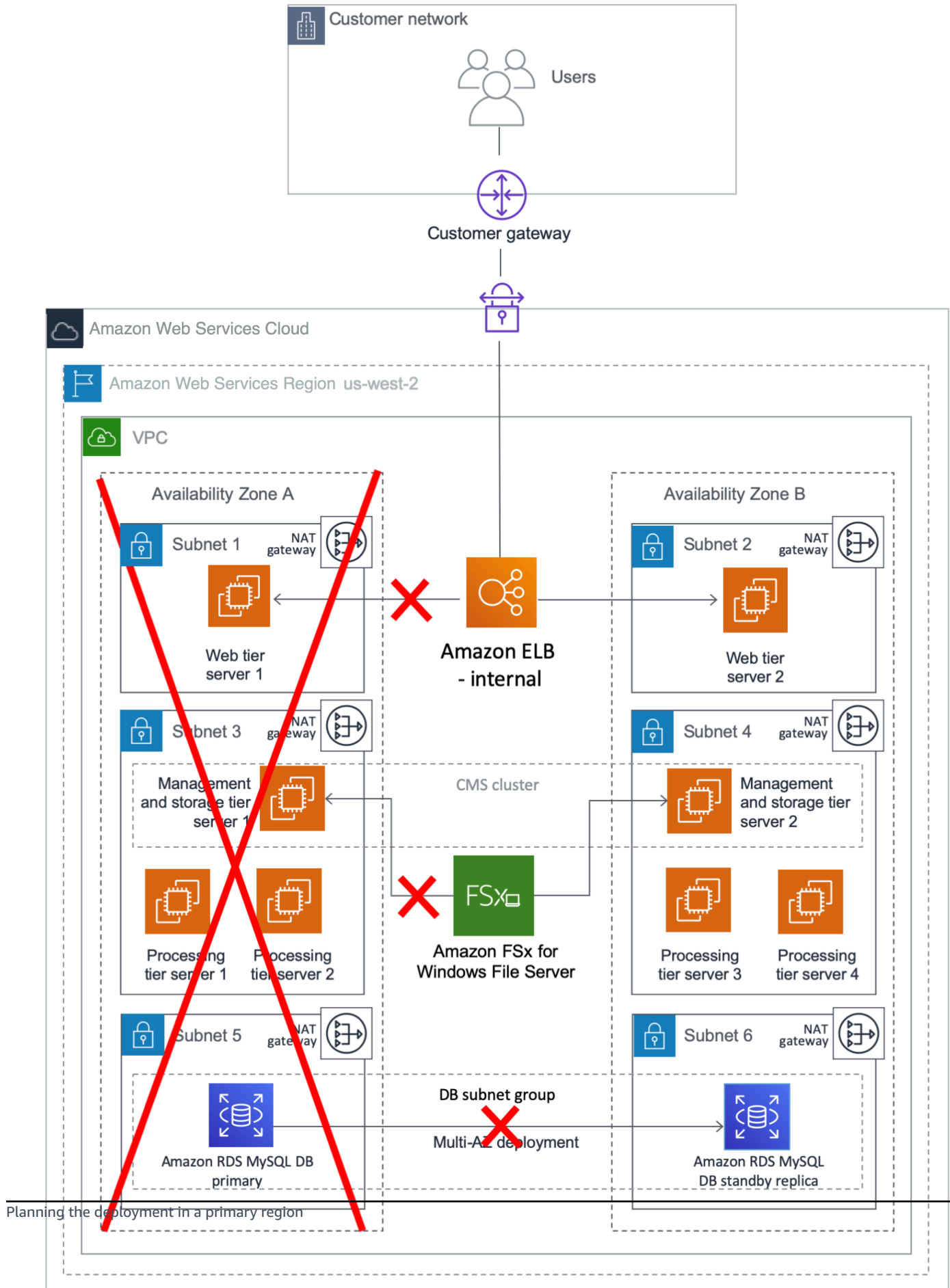


Figure 2: What happens during a failure in Availability Zone 1

Installing SAP BusinessObjects BI Platform for HA

Creating a VPC

1. Create a VPC in one of the available AWS Regions by following the instructions in the [Amazon Virtual Private Cloud documentation](#).
2. Specify a contiguous IP address range in CIDR block format (e.g., 192.168.0.0/16). You must choose a range that doesn't overlap with an already existing range used internally on your corporate network.
3. Create six new subnets, associate each with the new VPC, and split them across two different Availability Zones.
4. To continue with the installation process, you'll need access to the instances deployed into the VPC. You can use a [VPN connection](#) or [AWS Direct Connect](#) to establish a dedicated network connection from your premises to AWS. In both cases, be sure to create appropriate route tables, associate them with the new subnets, and adjust the network ACLs with rules to meet internal security requirements.

Note

The procedure in this section is based on the large-scale distributed deployment example. You can revise the procedure to support your architecture. For example, you can use additional subnets combined with network ACLs to further isolate or restrict access to different tiers of the SAP BusinessObjects BI Platform environment, and you can create multiple public and private subnets for isolation. Any subnet that is associated with an internet gateway will become a public subnet.

Building the Infrastructure

After you create the VPC, deploy supporting infrastructure instances that will provide key services used by the SAP environment from within the VPC. These services might include:

- Active Directory services (see [Quick Start](#))
- DNS
- Network address translation (NAT) services

- Remote Desktop gateways (see [Quick Start](#))
- Bastion hosts (see [Quick Start](#))

When supporting infrastructure services are in place, you can continue with the next section to start deploying SAP components into the VPC.

Preparing and Installing the CMS Database

This guide uses Amazon RDS MySQL for the CMS database. You can create a separate database for the auditing database if it's required. HA isn't a requirement for the auditing database, so this guide doesn't cover the steps for provisioning it.

1. Create a DB subnet group for an RDS instance by following the instructions in the [Amazon RDS documentation](#). Use two of the subnets you created earlier to create a DB subnet group.
2. In the [Amazon RDS console](#), launch an Amazon RDS MySQL DB instance by following the instructions in the [Amazon RDS documentation](#).
 - Choose a supported DB version based on the [Product Availability Matrix](#) on the SAP website, and select the instance type and storage based on your sizing output.
 - On the **Specify DB details** page, in the **Instance specifications** section, choose **Create replica in different zone**.
 - The **Choose use case** page asks if you are planning to use the DB instance you are creating for production. If you choose **Production - MySQL**, the Multi-AZ failover option is pre-selected.
 - On the **Configure advanced settings** page, provide information about the infrastructure you already provisioned, such as settings for the VPC, DB subnet group, and security group. In addition, you can provide custom options for encryption, backup retention period, maintenance window, and so on. You will also create a user to administer this database.
 - For the database name, you can provide the name you want to use for the CMS database. You can also change the database port from the default value of 3306 to your choice of port.
 - Choose **Create database**, and then wait for the DB instance status to change to **available** in the Amazon RDS console.
 - Choose the **Instances** view and note the **Endpoint** name. In case of failover to another Availability Zone, this endpoint enables an application to reconnect to a new primary database instance without having to change anything.

- (Optional) Create a CNAME in Amazon Route 53 for the database cluster endpoint. Use this CNAME during the installation of SAP BusinessObjects BI Platform nodes.

If you're planning a different database than Amazon RDS MySQL, use the database failover solution appropriate for your use case. Some solutions provide automatic failover on AWS in the event of primary database failure, including the failover of the virtual IP address used for database access. This method provides seamless failover for your applications.

Preparing the OS

Follow these steps to build and configure EC2 instances before installing SAP BusinessObjects BI Platform:

1. Launch a new Amazon EC2 Windows instance in Availability Zone 1. Prepare the server for the installation of SAP BusinessObjects BI Platform by following the instructions in the [SAP BusinessObjects BI Platform](#) installation guide.
2. Create an EBS volume and attach it to the EC2 Windows instance. For instructions, see the [Amazon EBS user guide](#).
3. Create and mount a file system by following the instructions in the [Amazon EBS user guide](#). Instructions include enabling automatic mounting each time you reboot the Windows instance.
4. Optionally, you can create an AMI of the instance, as described in the [AWS Toolkit for Visual Studio user guide](#), and launch all other SAP BusinessObjects BI Platform nodes by using this AMI.
5. Launch other EC2 instances according to your design, using one of these methods:
 - Manually perform steps 1-3 for all EC2 instances required by your architecture.
 - Create an AWS CloudFormation template, using the AMI you created in step 4 as a resource. Use this template to launch all additional instances in their respective Availability Zones. This option saves time and helps prevent manual errors during instance provisioning.

Installing CMS Nodes

1. Log in to each CMS server in the SAP BusinessObjects BI Platform node.
2. Launch the installation as described in the [SAP BusinessObjects BI Platform installation guide](#) and choose:

Custom / Expand > Expand an existing SAP BusinessObjects BI platform deployment > Instances > Servers > Platform Services

3. For the first CMS server installation, choose **Start a new SAP BusinessObjects BI platform deployment**.
4. For all additional CMS server installations, choose **Expand an existing SAP BusinessObjects BI platform deployment**.
5. Enter the details of the first CMS server when prompted. This will cluster the additional CMS servers with the first one.

Note

You can also perform a silent installation by using a response file and the procedure described in the SAP installation guide. This method enables you to quickly rerun the installation again on the same server or on another server, as required. You can further automate the installation by using an AWS CloudFormation template. The template will enable you to provision EC2 instances and install the BusinessObjects BI Platform application in just a few clicks.

Installing the Processing Tier

1. Log in to the processing tier server in the SAP BusinessObjects BI Platform node.
2. Launch the installation as described in the [SAP BusinessObjects BI Platform installation guide](#) and choose:

Custom / Expand > Expand an existing SAP BusinessObjects BI platform deployment > Instances > Servers

3. Choose the servers and other features that you want to install on the processing tier node, and run the installation.
4. Repeat these steps for all your processing tier servers.

Installing the Web Tier

1. Log in to each web tier server in the SAP BusinessObjects BI Platform node.

2. Launch the installation as described in the [SAP BusinessObjects BI Platform installation guide](#) and choose:

Custom / Expand > Expand an existing SAP BusinessObjects BI platform deployment > Instances > WebTier
3. Create a response file for silent installation by choosing **Custom / Expand > Web Tier**. You can install an integrated Tomcat web server by using the SAP installer, or you can use another supported web server and install applications by using the SAP installer.

Post-Installation Configuration for HA

When you have installed the SAP BusinessObjects BI Platform on all servers, you must configure the cluster name, Amazon FSx, and the load balancer to complete the HA configuration.

Configuring the SAP BusinessObjects BI Platform Cluster Name

Set up a cluster name by following the instructions in the [SAP BusinessObjects BI Platform Administrator Guide](#). Use this cluster name in the properties files of all web applications and client tools.

Configuring Amazon FSx for Input and Output Filestores

Note

Amazon FSx isn't available yet in all AWS Regions; see a [list of available regions](#). If you're planning to use a region that doesn't support Amazon FSx, you can choose a partner NFS solution from AWS Marketplace (such as Cloud Volumes ONTAP or SoftNAS) or you can deploy your own solution.

By default, the [File Repository Server \(FRS\)](#) filestore for SAP BusinessObjects BI Platform is located in the local installation directory. You can set up the filestore on a shared storage system like Amazon EFS, and then map your file share to all Windows EC2 instances that are configured to run storage tier servers:

1. Create and configure Amazon FSx by following the instructions provided at [Getting Started with Amazon FSx](#).
2. Configure Highly Available Amazon FSx by using instructions from [Deploying Multi-AZ File Systems](#).

3. Follow the instructions in the [SAP Community Wiki](#) to change the FRS path to Amazon FSx.

Configuring the Load Balancer for Web Server Access

To distribute the user load evenly across the web tier servers, you can use a load balancer between the web users and the web servers. In this guide, we'll discuss the use of [Elastic Load Balancing \(ELB\)](#) for this purpose. An Application Load Balancer automatically scales its request handling capacity in response to incoming application traffic. Follow these steps to configure an Application Load Balancer for SAP BusinessObjects BI Platform:

1. Enable Secure Sockets Layer (SSL) communications for SAP BusinessObjects BI Platform by following the instructions in the [administrator guide](#). See also: [Enabling SSL in BI Platform 4.2 SP05](#) on the SAP blog.
2. In the [Amazon EC2 console](#), [create an Application Load Balancer](#) in the VPC where SAP BusinessObjects BI Platform is running. Specify the Availability Zones and subnets of all the web tier servers.
3. [Configure a security group](#) that allows users to connect to the Application Load Balancer on the SSL port.
4. [Create a target group](#) to register web servers as the targets to the load balancer. For **Target type**, choose **ip** and specify the IP address and SSL port of the web servers to register as targets.
5. [Enable sticky sessions](#).
6. (Optional) Create a CNAME in Route 53 for the Application Load Balancer DNS name. Use this CNAME to access SAP BusinessObjects BI Platform.

HA for SAP Data Services

The HA architecture of SAP Data Services is very similar to the HA architecture of SAP BusinessObjects BI Platform. Before you can install SAP Data Services, you must have a working CMS server for SAP BusinessObjects BI Platform. You can install SAP Data Services on existing SAP BusinessObjects BI Platform nodes. If you do not have a SAP BusinessObjects BI Platform installation, SAP BusinessObjects Information Platform Services (IPS) can provide the basic CMS functions required by SAP Data Services. For either option, you can follow the steps in the previous sections to install the CMS server (by clustering two IPS CMS servers together), the CMS database, and a filestore with HA. You must use two Availability Zones for the CMS servers, a highly

available shared file system like Amazon FSx, and a CMS database with a Read Replica in a separate Availability Zone. Here are the high-level steps:

1. Create databases for the SAP Data Services repository in Amazon RDS MySQL. You can use your existing SAP BusinessObjects BI Platform or SAP BusinessObjects IPS CMS database engine to create a separate database for the repository. Alternatively, you can create a separate Amazon RDS MySQL instance for the repository.
2. Install SAP BusinessObjects BI Platform or SAP BusinessObjects IPS with a minimum of two CMS servers in two separate Availability Zones. See the [SAP Data Services installation guide](#) for IPS installation instructions.
3. Configure the filestore shared file system as described in [Configuring Amazon FSx for Input and Output Filestores](#).
4. Install SAP Data Services on all SAP BusinessObjects BI Platform servers or IPS servers.
5. Create an SAP Data Services central repository in the database created in step 1 by following the steps in the section *Repository management* in the [SAP Data Services administrator guide](#).
6. Configure HA for SAP Data Services batch jobs by following the instructions in [SAP Note 1938068](#).

HA Testing

[SAP Note 1229417](#) provides four tests you must perform to ensure that the CMSs are clustered correctly and that cluster functionality is working. Additionally, when performing the tests, you must also ensure that your deployment can tolerate Availability Zone failure. You can confirm this by stopping all the servers in one Availability Zone and making sure that all the services are still available. To test the CMS database, you can perform the failover manually. In the [Amazon RDS console](#), choose **Instances**, **Instance actions**, **Failover**. After all servers in one Availability Zone are shut down and database failover is complete, test the availability of the system and all the services. Repeat the same test for the second Availability Zone.

Disaster Recovery

The DR approach you take for SAP BusinessObjects BI Platform, as for any other enterprise application, depends on your RTO and RPO requirements. As discussed in [SAP Note 2056228](#), there are two options for building a DR site for SAP BusinessObjects BI Platform:

- Fully or selectively using SAP Lifecycle Manager (LCM) or Data Federation Services to promote or distribute the content from the primary system.
- Periodically copying over the CMS database and FRS contents, and using that to start a secondary system when required.

DR for SAP BusinessObjects BI Platform on AWS

DR for SAP BusinessObjects BI Platform on the AWS Cloud refers to a scenario in which the primary AWS Region where SAP BusinessObjects BI Platform application is running is unavailable. The goal of setting up a DR site is to be able to recover the application within your acceptable RTO and RPO.

There are no restrictions in AWS for using LCM or Data Federation Services for your DR environment. Note that using LCM requires either consuming resources on the source system or provisioning an additional system to run promotion management jobs. This option might also result in a higher RPO depending on the frequency of promotion management jobs. See [Promotion Management Architecture](#) on the SAP Community Wiki for the high-level architecture for this option.

In this guide, we'll discuss the second option for handling DR, which is to copy the CMS database and FRS contents. Using variants of this option, you can build the complete primary system within your recovery time limits. This option doesn't require resources from the primary system except for the backup copy of the database and file system.

Planning the Deployment in the DR Region

Planning the deployment of a DR system is critical. Consider these items in your planning:

1. Choose a region for your DR site. Make sure that this DR region supports the services and resources that you are using for your SAP BusinessObjects BI Platform. Each AWS Region is completely isolated from other AWS Regions to provide the best possible fault tolerance and stability. You may choose a DR region that is geographically closer to your users. For the examples in this guide, we'll assume US West (Oregon) as the primary region and US East (N. Virginia) as the DR region.
2. Determine the RTO and RPO that are acceptable to your business. This decision determines the methodologies you choose for copying the CMS database and FRS content from the primary site to the DR site. See [Affordable Enterprise-Grade Disaster Recovery Using AWS](#) on the AWS website for options based on your RTO and RPO requirements. In this guide, because

we're using Amazon RDS MySQL for the CMS database, we have two options for copying the database:

- Perform periodic snapshots of the Amazon RDS MySQL database and copy these snapshots to the DR region (results in lower RTO and RPO, and lower cost).
- Create an Amazon RDS MySQL Read Replica in the DR region, and promote this Read Replica to master in the event of a disaster or DR drill (results in higher RTO and RPO, and higher cost).

For details, see the section [Installing and Configuring the CMS Database for the DR Region](#) later in this guide.

3. Determine the capacity of the DR site. You might want the DR site to have the same capacity and resources as the primary site, or you might decide to run it with fewer resources.
4. For the examples in this guide, we're using a Route 53 CNAME with an Application Load Balancer for user connectivity. You can change the value of the CNAME to point to the load balancer of the DR region in the event of a disaster or DR drill. This will direct users to the DR environment. You can also use other methods to redirect users if you aren't using Route 53.
5. Determine the failover process. Define the disaster situations beforehand and put a mechanism in place to direct users to the DR site if a disaster occurs. Define test cases to simulate these scenarios, and perform a DR drill.

Designing Network and Security Groups for the DR Region

Network and security groups in the DR region will follow the same design as in the primary region, as discussed in the section [Designing Network and Security Groups for the Primary Region](#) earlier in this guide. The table in that section lists the VPC and subnet CIDRs that the DR region uses for the examples in this guide. Figure 3 shows the DR architecture used in this guide. This guide does not discuss the use of a secondary Availability Zone for HA in the DR region, but based on your requirements, you can use one to install additional servers.

Multiple Region Multi-VPC Connectivity

See [Multiple Region Multi-VPC Connectivity](#) on the AWS website to find options for connecting multiple VPCs in different AWS Regions. If you aren't using an Amazon RDS database, you can use the database replication method supported by your database vendor to build a DR database and continuously copy the data. SAP HANA System Replication is one such example. AWS provides options to establish a connection between the primary and DR VPCs in two separate AWS Regions

securely and reliably. [Multiple Region Multi-VPC Connectivity](#) discusses two ways to connect the VPCs:

- **Routing over AWS networks:** This option enables you to use an AWS backbone network for connectivity. AWS provides a high bandwidth, global network infrastructure to support your regional networking needs. AWS Regions are connected to multiple Internet Service Providers (ISPs) and to a private global network backbone, which provides lower cost and more consistent cross-region network latency when compared with the public internet.
- **Routing over non-AWS networks:** This option enables you to route DR network traffic through your corporate data center. Customers can use an internet-based VPN or their own corporate network backbone.

Each option provides different benefits for bandwidth reliability, security, HA, and use of existing infrastructure, and might have limitations as well. You can choose the option that meets your requirements. You can also use AWS Direct Connect gateways to connect your AWS Direct Connect connection over a private virtual interface to one or more VPCs located in the same or different regions in your account. See [AWS Direct Connect Gateways](#) in the AWS documentation for the rules and limitations of using this approach.

 **Note**

The AWS Regions and Availability Zones shown in the following diagrams are just examples. This architecture can be used in any AWS Region.

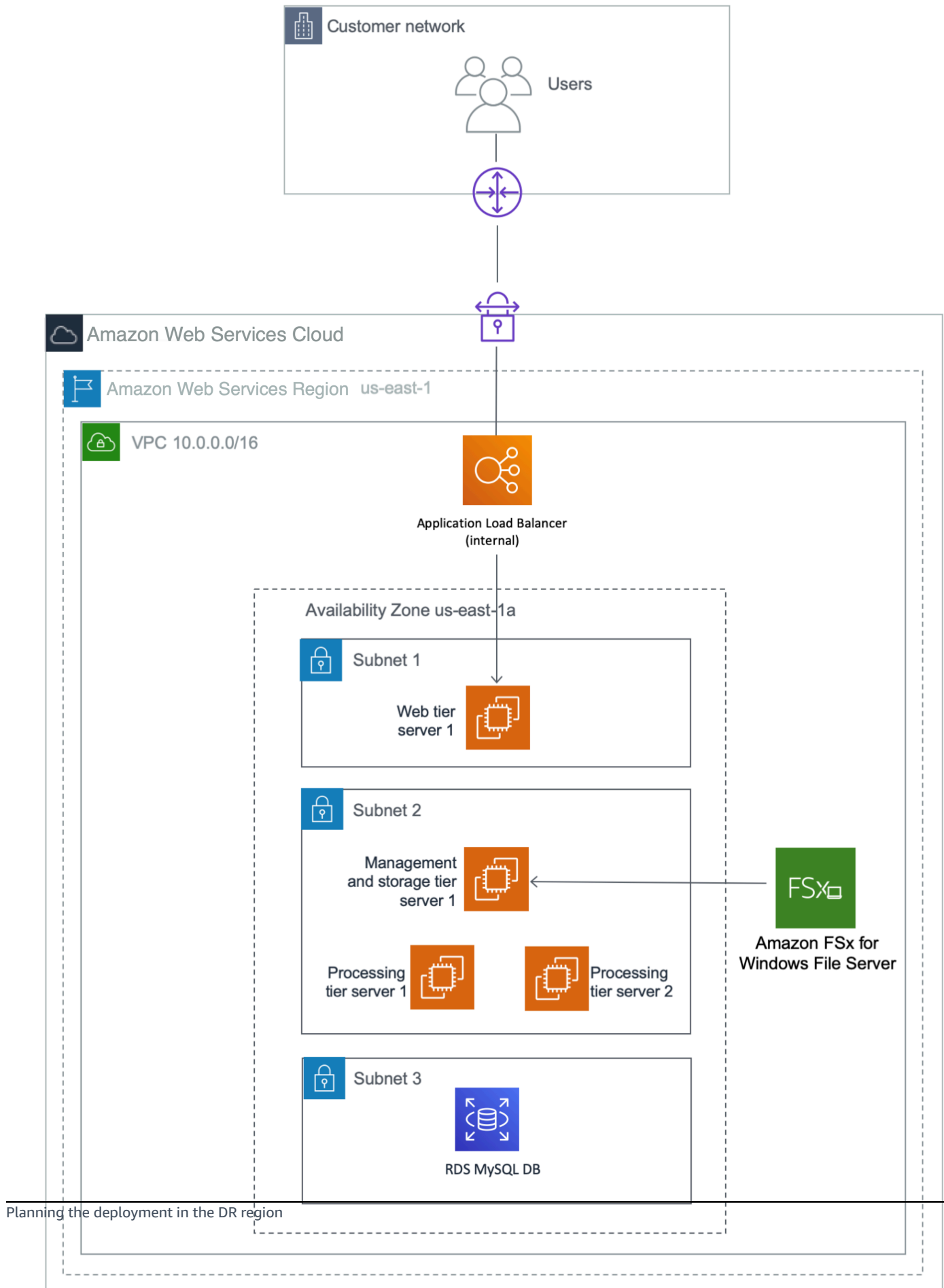


Figure 3: DR architecture for SAP BusinessObjects BI Platform

Figure 4 shows the user connectivity using Route 53 as an example. You can also use your own DNS. During normal operations, the value of the CNAME record for SAP BusinessObjects BI Platform points to the Application Load Balancer of the primary region. The load balancer distributes the load between the available web servers that deploy the SAP BusinessObjects BI Platform web applications.

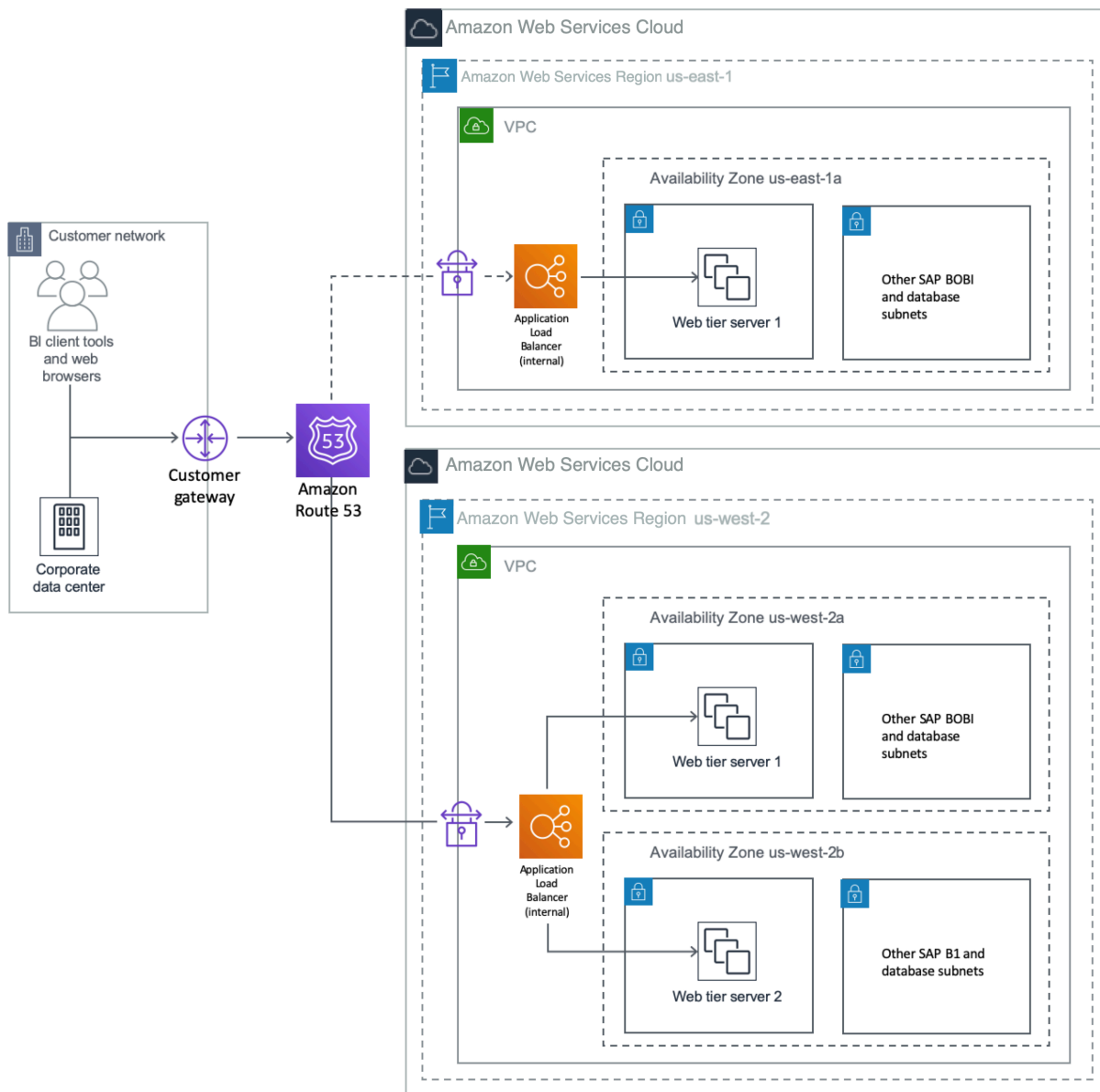


Figure 4: User connectivity with Route 53

In the event of DR or when performing a DR drill, you can either manually switch the DNS to point to the DR region or you can use automatic failover by using Route 53 health checks. For details, see [Configuring DNS Failover](#) in the Route 53 documentation. Redirect all the users to the DR region as shown in Figure 5.

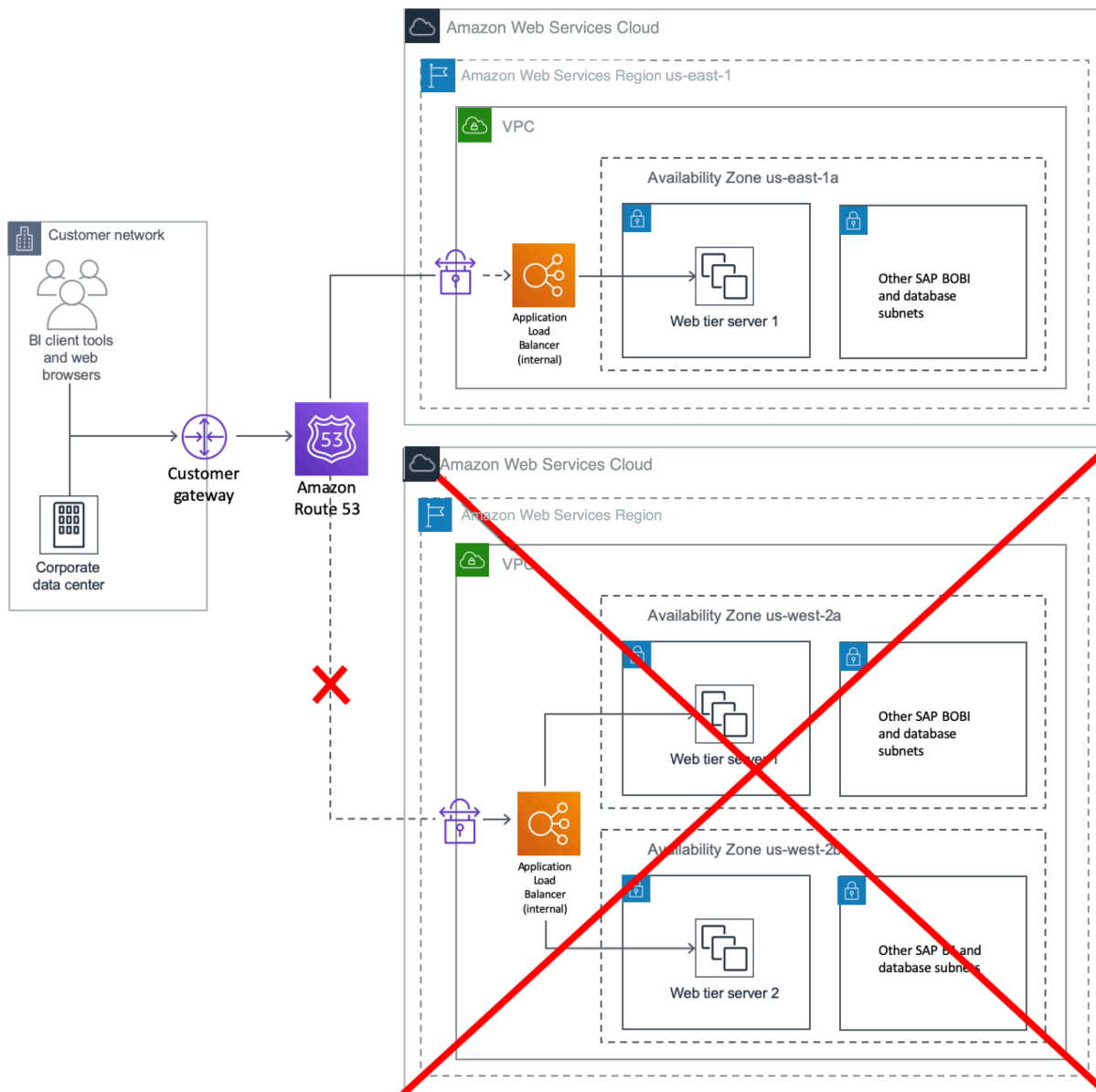


Figure 5: Redirecting users to a DR region

Installing SAP BusinessObjects BI Platform in the DR Region

When you have selected the DR region, follow the steps in the [Preparing the OS](#) section earlier in this guide to build SAP BusinessObjects BI Platform EC2 instances in that region.

Installing and Configuring the CMS Database for the DR Region

The CMS database in the DR region must be a copy of the CMS database in the primary region. The method for copying this database depends on the database type and your RTO and RPO requirements. This guide describes two options for an Amazon RDS MySQL database.

- Option 1: Create a cross-region Read Replica of the primary Amazon RDS MySQL database.

Amazon RDS uses the MySQL DB engine's built-in replication functionality to create a special type of DB instance called a Read Replica from a source DB instance. Updates made to the source DB instance are asynchronously copied to the Read Replica. Create a Read Replica of the primary CMS database by following the instructions in the [Amazon RDS documentation](#). The documentation also provides instructions for promoting a Read Replica to a primary DB instance. This option will provide a lower RPO and RTO than Option 2.

- Option 2: Copy primary database snapshots to the DR region and use them to build a DR CMS database.

Create a snapshot of the Amazon RDS MySQL DB and copy the snapshot to the DR region. Use this snapshot to create a new instance of the Amazon RDS MySQL DB, and use this instance to configure the SAP BusinessObjects BI Platform DR database. See the [Amazon RDS documentation](#) for instructions, features, and limitations of this option. For a continuous copy of the database content, you can create and schedule scripts to perform the snapshot and copy them to the DR region. You can also use open-source tools to deploy such solutions, but make sure to test any tools thoroughly before using them for your DR deployment. When your testing is complete and the primary region is active, you can delete the RDS instance. Create a new instance by using the latest snapshot when the DR system is required again.

Installing a CMS Node

The steps to install the SAP BusinessObjects BI Platform application in the DR region are described in [SAP Note 2603573](#). This procedure uses SQL Anywhere to temporarily create Server Intelligence Agents (SIAs) on all instances with a local CMS database. In the next section, we'll reconfigure all nodes to form a single SAP BusinessObjects BI Platform cluster.

1. Log in to all CMS servers for your SAP BusinessObjects BI Platform DR nodes, based on your design.
2. Install the CMS servers by following the instructions in the SAP Note and using the same SIA names as in the production cluster.

Starting SAP BusinessObjects BI Platform in the DR Region

Create the SIA nodes for the CMS servers by following the procedure described in [SAP Note 2603573](#). When prompted for the CMS database details, use the connection information and credentials of the database created in the previous section. The CMS servers should now start correctly. You can now install the remaining tiers, such as the processing tier or web tier, as required by your design. The installation instructions for the primary region also apply to the DR region.

Creating an Amazon FSx Filestore for the DR Region

The filestore for the DR region in the Amazon FSx file system must be a copy of the filestore in the primary region.

1. Stop the input and output file servers in the DR region and follow the instructions in [Configuring Amazon FSx for Input and Output Filestores](#) to create an Amazon FSx file system.
2. Prepare S3 buckets in both the primary and DR regions for copying Amazon FSx content.
3. Copy the filestore in Amazon FSx in the primary region to the S3 bucket in that region by using simple scripts and setting up a frequency that matches your database snapshots. The frequency depends on your RPO.
4. Enable cross-region replication from the bucket in the primary region to the DR region.
5. Before starting the SAP BI servers in the DR region, copy the contents of the bucket to the storage tier servers' Amazon FSx mapped directory for the filestore.
6. Start SAP BusinessObjects BI Platform applications in the DR region. You should now be able to verify that the SAP BusinessObjects BI Platform system in the DR region has all the data of the primary region, as of the time the database and filestore were last copied.

Performing the Failover

When you complete all the steps described in the previous sections, the SAP BusinessObjects BI Platform system will be up and running in the DR region with primary region data. You can

shut down the EC2 instances in the DR region when you don't need them. Monitor the copy of the database and Amazon FSx file system to make sure that they're active according to your configuration. In the event of a disaster or DR drill, follow these steps:

1. Start the CMS database in the DR region using one of these options:
 - If you're using a Read Replica, open the [Amazon RDS console](#) in the DR region, and then choose **Instances**. Select the DB instance that is the Read Replica of the production database, and then choose **Instance Actions, Promote Read Replica** to start the CMS database in the DR region.
 - If you're using snapshots, open the [Amazon RDS console](#) in the DR region, and then choose **Snapshots**. Select the latest snapshot and restore that snapshot to create a CMS database in the DR region.
2. If you're using snapshots, update the data source settings for SAP BusinessObjects BI Platform in the DR region by following the instructions in the section *To select a new or existing CMS database on UNIX* in the [SAP BusinessObjects BI Platform Administration Guide](#). Use the RDS instance created with the latest CMS database snapshot in the primary region.
3. Copy the contents of the S3 bucket that has the latest Amazon FSx content (from the primary region) to the Amazon FSx file system for the input and output filestore.
4. Start SAP BusinessObjects BI Platform in the DR region and perform validation.
5. Redirect your users by changing the value of the CNAME used for SAP BusinessObjects BI Platform from the Application Load Balancer for the primary region to the load balancer for the DR region.

DR for SAP Data Services

As described earlier in the [HA for SAP Data Services](#) section, you can install SAP Data Services on either an existing SAP BusinessObjects BI Platform node or on SAP BusinessObjects Information Platform Services (IPS). In both cases, the DR approach described previously in this section applies to SAP Data Services as well.

Summary

SAP customers can now deploy SAP BusinessObjects BI Platform solution and landscapes on the scalable, on-demand Amazon EC2 platform in a highly available manner without having to invest in costly capital expenditures for the underlying infrastructure. By combining the flexibility of the

AWS platform with SAP installation techniques, SAP customers can greatly improve the availability of their deployments. For further information, including case studies of customers who have deployed SAP systems on AWS, see [SAP Case Studies](#) on the AWS website.

Additional Tips

Tagging AWS Resources

Adding tags to AWS objects will make it much easier to manage the SAP HA environment, and can also help you search for resources quickly. You can use Amazon EC2 API calls in conjunction with a special tag filter. For more information about tagging resources, see the [AWS documentation](#).

Third-Party Software Components

Additional third-party components might be integral to running business processes within an SAP environment. After you determine your requirements, consider leveraging some of the concepts discussed in this guide, such as:

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