AWS Launch Wizard: User Guide
Copyright © 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

AWS Launch Wizard for Active Directory .............................................................. 1  
What is AWS Launch Wizard for Active Directory? .............................................. 1  
  Supported versions .......................................................................................... 1  
  Features ........................................................................................................... 2  
  Components .................................................................................................... 3  
  Requirements .................................................................................................. 3  
  Related services .............................................................................................. 4  
  How it works .................................................................................................... 5  
  Domain controller launch limits ..................................................................... 10  
  AWS Regions .................................................................................................. 10  
Get started ......................................................................................................... 10  
  Access ............................................................................................................. 11  
  Set Up ............................................................................................................ 11  
  Deploy ............................................................................................................. 15  
  Configure trusts ............................................................................................. 20  
Manage application resources ......................................................................... 20  
Best practices ..................................................................................................... 21  
  High availability ............................................................................................. 21  
  Security in Launch Wizard for Active Directory ............................................. 21  
Troubleshoot ..................................................................................................... 22  
  Launch Wizard provisioning events ............................................................... 23  
  CloudWatch Logs ......................................................................................... 23  
  AWS CloudFormation stack .......................................................................... 23  
  Errors ............................................................................................................. 23  
AWS Launch Wizard for Amazon Elastic Kubernetes Service ......................... 24  
What is AWS Launch Wizard for Amazon Elastic Kubernetes Service? ............ 24  
  Deployment options ....................................................................................... 24  
  Components .................................................................................................... 24  
Get Started ....................................................................................................... 26  
  Access ............................................................................................................. 26  
  Set up ............................................................................................................. 26  
  Deploy to a new VPC .................................................................................... 27  
  Deploy to an existing VPC ............................................................................ 33  
Test the deployment ......................................................................................... 40  
Best practices ..................................................................................................... 40  
  Amazon EKS application best practices ....................................................... 40  
  Use AWS CloudFormation for ongoing management ............................... 41  
  Monitor additional resource usage ............................................................... 41  
  Security .......................................................................................................... 41  
Troubleshoot ..................................................................................................... 41  
  Launch Wizard provisioning events ............................................................... 41  
  AWS CloudFormation stack .......................................................................... 41  
  Application launch quotas ............................................................................ 42  
  Enable termination protection ...................................................................... 42  
  Errors ............................................................................................................. 43  
AWS Launch Wizard for Remote Desktop Gateway ......................................... 44  
What is AWS Launch Wizard for Remote Desktop Gateway? .......................... 44  
  Deployment options ....................................................................................... 44  
  ......................................................................................................................... 45  
Get Started ....................................................................................................... 47  
  Access ............................................................................................................. 48  
  Set up ............................................................................................................. 48  
  Deploy to a new VPC .................................................................................... 49
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Launch Wizard for SQL Server</td>
<td>153</td>
</tr>
<tr>
<td>What Is AWS Launch Wizard for SQL Server?</td>
<td>153</td>
</tr>
<tr>
<td>Supported versions</td>
<td>154</td>
</tr>
<tr>
<td>Features</td>
<td>154</td>
</tr>
<tr>
<td>Components (Windows)</td>
<td>157</td>
</tr>
<tr>
<td>Components (Linux)</td>
<td>159</td>
</tr>
<tr>
<td>Related services</td>
<td>160</td>
</tr>
<tr>
<td>How it works</td>
<td>161</td>
</tr>
<tr>
<td>Deployment options</td>
<td>165</td>
</tr>
<tr>
<td>Default quotas</td>
<td>167</td>
</tr>
<tr>
<td>Get started</td>
<td>167</td>
</tr>
<tr>
<td>Set Up</td>
<td>167</td>
</tr>
<tr>
<td>Deploy on Windows</td>
<td>173</td>
</tr>
<tr>
<td>Deploy on Ubuntu</td>
<td>185</td>
</tr>
<tr>
<td>Deploy on RHEL</td>
<td>192</td>
</tr>
<tr>
<td>Manage application resources with Launch Wizard for SQL Server</td>
<td>198</td>
</tr>
<tr>
<td>Manage application resources with SSM Application Manager</td>
<td>199</td>
</tr>
<tr>
<td>Use runbooks</td>
<td>200</td>
</tr>
<tr>
<td>Onboard existing applications</td>
<td>201</td>
</tr>
<tr>
<td>Patch management</td>
<td>203</td>
</tr>
<tr>
<td>Automation documents</td>
<td>203</td>
</tr>
<tr>
<td>AWSSQLServer-DBCC</td>
<td>203</td>
</tr>
<tr>
<td>AWSSQLServer-Backup</td>
<td>204</td>
</tr>
<tr>
<td>AWSSQLServer-Index</td>
<td>204</td>
</tr>
<tr>
<td>AWSSQLServer-Restore</td>
<td>204</td>
</tr>
<tr>
<td>Monitoring</td>
<td>205</td>
</tr>
<tr>
<td>Best practices</td>
<td>205</td>
</tr>
<tr>
<td>High availability</td>
<td>205</td>
</tr>
<tr>
<td>Automatic failover</td>
<td>206</td>
</tr>
<tr>
<td>Security groups and firewalls</td>
<td>206</td>
</tr>
<tr>
<td>Troubleshoot</td>
<td>207</td>
</tr>
<tr>
<td>Active Directory objects and DNS record clean up (deployment on Windows)</td>
<td>207</td>
</tr>
<tr>
<td>Launch Wizard provisioning events</td>
<td>208</td>
</tr>
<tr>
<td>CloudWatch Logs</td>
<td>208</td>
</tr>
<tr>
<td>SSM Automation execution</td>
<td>208</td>
</tr>
<tr>
<td>AWS CloudFormation stack</td>
<td>209</td>
</tr>
<tr>
<td>Pacemaker on Ubuntu (deployment on Linux)</td>
<td>209</td>
</tr>
<tr>
<td>SQL Server Management Studio</td>
<td>210</td>
</tr>
<tr>
<td>Errors</td>
<td>210</td>
</tr>
<tr>
<td>Security</td>
<td>212</td>
</tr>
<tr>
<td>Infrastructure Security</td>
<td>212</td>
</tr>
<tr>
<td>Resilience</td>
<td>212</td>
</tr>
<tr>
<td>Data Protection</td>
<td>213</td>
</tr>
<tr>
<td>Encryption with AWS managed keys and customer managed keys</td>
<td>213</td>
</tr>
<tr>
<td>Identity and Access Management</td>
<td>214</td>
</tr>
<tr>
<td>Update Management</td>
<td>215</td>
</tr>
<tr>
<td>AWS managed policies</td>
<td>215</td>
</tr>
<tr>
<td>AmazonLaunchWizard_Fullaccess</td>
<td>216</td>
</tr>
<tr>
<td>AmazonEC2RolePolicyForLaunchWizard</td>
<td>226</td>
</tr>
</tbody>
</table>
Policy updates ................................................................................................................ 230
Document History .................................................................................................................. 234
AWS Launch Wizard for Active Directory

This section of the AWS Launch Wizard documentation provides guidance for deploying self-managed domain controllers using the Launch Wizard service.

Topics

- What is AWS Launch Wizard for Active Directory? (p. 1)
- Get started with AWS Launch Wizard for Active Directory (p. 10)
- Manage application resources with AWS Launch Wizard for Active Directory (p. 20)
- High availability and security best practices for AWS Launch Wizard for Active Directory (p. 21)
- Troubleshoot AWS Launch Wizard for Active Directory (p. 22)

What is AWS Launch Wizard for Active Directory?

AWS Launch Wizard for Active Directory is a service that applies AWS cloud application best practices to guide you through setting up a new Active Directory infrastructure, or adding domain controllers to an existing infrastructure, either in the AWS Cloud or on premises. The deployment environment includes existing VPCs, security groups, and AWS Identity and Access Management (IAM) roles. You can set up a new Active Directory infrastructure with two to three domain controllers, or you can add up to three domain controllers to your existing Active Directory infrastructure for each AWS Region.

Launch Wizard reduces the time that it takes to set up an Active Directory infrastructure and deploy self-managed domain controllers to the cloud or on premises. You input your domain controller requirements, including number of nodes and connectivity, on the service console, and AWS Launch Wizard identifies the right AWS resources to deploy your self-managed domain controllers. AWS Launch Wizard provides an estimated cost of deployment, and gives you the ability to modify your resources and instantly view the updated cost assessment. When you approve, AWS Launch Wizard provisions and configurations the selected resources in a few hours to create fully-functioning, production-ready domain controllers.

After you deploy your self-managed domain controllers, they are ready to use and can be accessed on the Amazon Elastic Compute Cloud (Amazon EC2) console.

Contents

- Supported operating systems (p. 1)
- Features of AWS Launch Wizard (p. 2)
- Components (p. 3)
- Requirements (p. 3)
- Related services (p. 4)
- How AWS Launch Wizard works (p. 5)
- Domain controller launch limits (p. 10)
- AWS Regions (p. 10)

Supported operating systems

AWS Launch Wizard supports the following operating systems:
Features of AWS Launch Wizard

**AWS Launch Wizard provides the following features:**
- Simple application deployment (p. 2)
- AWS resource selection (p. 2)
- Cost estimation (p. 2)
- SNS notification (p. 2)
- Early input validation (p. 2)
- Application resource groups for easy discoverability (p. 3)

**Simple application deployment**

AWS Launch Wizard makes it efficient for you to deploy self-managed domain controllers on AWS. When you enter the domain controller requirements, AWS Launch Wizard deploys the necessary AWS resources for a production-ready environment. This means that you do not have to manage separate infrastructure pieces or spend time provisioning and configuring your domain controllers.

**AWS resource selection**

Launch Wizard considers the number of Active Directory users to determine the best instance type, EBS volumes, and other resources for your domain controllers. You can modify the recommended defaults.

**Cost estimation**

Launch Wizard provides a cost estimate for the complete deployment that is itemized for each individual resource being deployed. The estimated cost automatically updates each time you change a resource type configuration in the wizard. However, the provided estimates are only for general comparisons. They are based on on-Demand costs and actual costs may be lower.

**SNS notification**

You can provide an SNS topic that allows Launch Wizard to send you notifications and alerts about the status of a deployment.

**Early input validation**

You can take advantage of your existing infrastructure, such as VPC or security groups, with Launch Wizard. This may lead to deployment failures if your existing infrastructure does not meet certain deployment prerequisites. If these requirements are not met, the deployment will fail. If you are in a later stage of a deployment, this failure can take more than an hour to detect. To detect these types of issues early in the application deployment process, Launch Wizard's validation framework verifies key infrastructure specifications before provisioning. Verification takes approximately 15 minutes. If necessary, you can take appropriate actions to adjust your VPC configuration.

**Note**

Some validations, such as for Active Directory credentials, require Application Wizard to launch a t2.large EC2 instance in your account for a few minutes. After it runs the necessary validations, Launch Wizard terminates the instance.
Application resource groups for easy discoverability

Launch Wizard creates a resource group for all of the AWS resources created for your domain controllers. You can manage the resources through the Amazon EC2 console or with Systems Manager. When you access Systems Manager through Launch Wizard, the resources are automatically filtered for you based on your resource group.

Components

Self-managed domain controllers deployed with Launch Wizard include the following components:

- A virtual private cloud (VPC) configured with public and private subnets across two Availability Zones. A public subnet is a subnet whose traffic is routed to an internet gateway. If a subnet does not have a route to the internet gateway, then it is a private subnet. The VPC provides the network infrastructure for your domain controller environment.
- Amazon EC2 instances on which to provision your domain controllers.
- An internet gateway to provide access to the internet.
- In the public subnets, network address translation (NAT) gateways for outbound internet access. If you are deploying in your preexisting VPC, Launch Wizard uses the existing NAT gateway in your VPC. For more information about NAT gateways, see NAT Gateways.
- Elastic IP addresses associated with the NAT gateway and RDGW instances. For more information about Elastic IP addresses, see Elastic IP Addresses.
- AWS CloudFormation templates and PowerShell configuration scripts to perform the domain controller configuration steps.
- Security groups to ensure the secure flow of traffic between the instances deployed in the VPC. For more information, see Security Groups for Your VPC.
- AWS Secrets Manager to protect secrets required to generate and store your Active Directory Administrator credentials.
- Amazon CloudWatch Logs to monitor, store, and access your log files produced by AWS CloudFormation.

Requirements

Your account must be configured as specified in the following table to deploy self-managed domain controllers using Launch Wizard.

To add domain controllers to an existing infrastructure, you must create a VPC peering connection between the two VPCs for an existing Active Directory in AWS. If you are using an existing Active Directory on premises, you must use AWS Direct Connect. To ensure that instances in the VPCs can communicate with each other, you can use either Direct Connect or VPC Private Link. For more information about VPC connectivity, see VPN connections.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Minimum number of resources required for deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual private clouds (VPCs)</td>
<td>1</td>
</tr>
<tr>
<td>VPC security groups</td>
<td>3</td>
</tr>
<tr>
<td>AWS Identity and Access Management (IAM) roles</td>
<td>2</td>
</tr>
<tr>
<td>General purpose EC2 instances</td>
<td>Existing VPC: 1</td>
</tr>
</tbody>
</table>
### Related services

<table>
<thead>
<tr>
<th>Resource</th>
<th>Minimum number of resources required for deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Active Directory infrastructure</td>
<td>2</td>
</tr>
<tr>
<td>AWS Secrets Manager secrets</td>
<td>2</td>
</tr>
</tbody>
</table>

If you have an existing environment that uses these resources and you think that deploying domain controllers in this environment using Launch Wizard may exceed your default quotas, you can request service quota increases for these resources. For default quotas, see AWS service quotas.

For additional prerequisites to deploy domain controllers using Launch Wizard, see Set up for AWS Launch Wizard for Active Directory (p. 11).

### AWS CloudFormation

AWS CloudFormation is a service for modeling and setting up your AWS resources, enabling you to spend more time focusing on your applications that run in AWS. You create a template that describes all of the AWS resources that you want to use (for example, EC2 instances), and AWS CloudFormation provisions and configures those resources for you. With Launch Wizard, you don’t have to sift through CloudFormation templates to deploy your application. Instead, Launch Wizard combines infrastructure provisioning and configuration (with an AWS CloudFormation template and PowerShell scripts) to provision a new Active Directory infrastructure or additional domain controllers in your account. For more information, see the AWS CloudFormation User Guide.

### Amazon Simple Notification Service (SNS)

Amazon Simple Notification Service (Amazon SNS) is a highly available, durable, secure, fully managed publish/subscribe messaging service that provides topics for high-throughput, push-based, many-to-many messaging. Using Amazon SNS topics, your publisher systems can fan out messages to a large number of subscriber endpoints and send notifications to end users using mobile push, SMS, and email. You can use Amazon SNS topics for your Launch Wizard deployments to stay up to date on deployment progress. For more information, see the Amazon Simple Notification Service Developer Guide.

### Amazon CloudWatch Logs

Amazon CloudWatch Logs enables you to centralize the logs from all of your systems, applications, and AWS services that you use, in a single, highly scalable service. You can then easily view them, search them for specific error codes or patterns, filter them based on specific fields, or archive them securely for future analysis. Amazon CloudWatch Logs enables you to see all of your logs, regardless of their source, as a single and consistent flow of events ordered by time, and you can query them and sort them based on other dimensions, group them by specific fields, create custom computations with a powerful query language, and visualize log data in dashboards. Launch Wizard streams provisioning logs from all of the AWS log sources that you can view on the CloudWatch console.
AWS Secrets Manager

With AWS Secrets Manager you can replace hard-coded credentials in your code, including passwords, with an API call to Secrets Manager to programmatically retrieve the secret. This helps ensure the secret can't be compromised by someone examining your code. Also, you can configure Secrets Manager to automatically rotate the secret for you according to a specified schedule. Launch Wizard uses Secrets Manager to join your domain controllers to Active Directory and promote them.

How AWS Launch Wizard works

AWS Launch Wizard provides a complete solution to provision self-managed domain controllers on AWS. You select Active Directory in the wizard and provide the specifications, such as number of users. Based on the infrastructure requirements that you enter, Launch Wizard automatically provisions the right AWS resources in the cloud. For example, Launch Wizard determines the best instance type and EBS volume for your number of users, then deploys and configures them.

Launch Wizard provides an estimated cost of deployment. You can modify your resources and instantly view an updated cost assessment. Once you approve, Launch Wizard validates the inputs and flags inconsistencies. After you resolve the inconsistencies, Launch Wizard provisions the resources and configures them. The result is a ready-to-use Active Directory infrastructure and domain controllers.

AWS Launch Wizard performs the following tasks to provision self-managed domain controllers.

• Sets up the VPC, including private and public subnets in two Availability Zones.*
• Configures two NAT gateways in the public subnets.*
• Configures private and public routes.*
• Enables ingress traffic into the VPC for administrative access to Remote Desktop Gateway.
• Stores the trust administrator password in Secrets Manager.
• Uses Secrets Manager to generate Domain Administrator passwords.
• Launches instances using the specified version of Windows Server.
• Configures security groups and rules for traffic between instances.
• Sets up and configures Active Directory sites and subnets.
• For existing VPCs, optionally sets up forest trusts with other Active Directory forests. For the required prerequisites to set up forest trusts, see Trust relationships (p. 12). For information about creating forest trusts, see Configure forest trust relationships (p. 20).
• Sets up and deploys Active Directory Certificate Services with a new Active Directory infrastructure.
• For existing VPCs, adds up to three domain controllers to an existing Active Directory infrastructure. For the required prerequisites to add domain controllers, see Set up for AWS Launch Wizard for Active Directory (p. 11).

* If you deploy Launch Wizard into an existing VPC, the tasks in this list marked by asterisks are skipped.

Topics

• Deployment path (p. 5)
• Implementation details (p. 6)

Deployment path

Launch Wizard supports the following deployment path for provisioning self-managed domain controllers.
Deploy and manage your own domain controllers on Amazon EC2 instances

Launch Wizard builds the AWS Cloud infrastructure, and sets up and configures Active Directory Domain Services (AD DS) and Active Directory-integrated DNS on the AWS Cloud. It does not include AWS Directory Service, so you handle all AD DS maintenance and monitoring tasks. You can deploy the domain controllers into a new or existing VPC infrastructure.

Implementation details

This section describes how Launch Wizard implements an Active Directory Domain Services (AD DS) deployment in the AWS Cloud. It includes details about how to use Amazon VPC to define your networks in the cloud, and information about domain controller placement, Active Directory Sites and Services configuration, and how DNS and DHCP work in an Amazon Virtual Private Cloud (Amazon VPC).

Topics
- VPC (p. 6)
- Security groups (p. 7)
- Remote Desktop Gateway (p. 7)
- Active Directory (p. 7)
- Self-managed domain controller architecture (p. 10)

VPC

You can define a virtual network topology that closely resembles a traditional on-premises network using Amazon VPC. A VPC can span multiple Availability Zones place independent infrastructure in physically separate locations. A multi-Availability Zone deployment results in high availability and fault tolerance. Launch Wizard provisions domain controllers in two Availability Zones to provide highly available, low latency access to AD DS services in the AWS Cloud.

Launch Wizard can build a new VPC for the deployment, or deploy into an existing VPC. To accommodate highly available AD DS in the AWS Cloud, Launch Wizard builds (or requires, in the case of existing VPCs) a base Amazon VPC configuration that complies with the following AWS best practices:

- Domain controllers must be placed in a minimum of two Availability Zones to provide high availability.
- Domain controllers and other non-internet facing servers must be placed in private subnets.
- Launched instances require internet access to connect to the AWS CloudFormation endpoint during the bootstrapping process. To support this configuration, public subnets are used to host NAT gateways for outbound internet access. Remote Desktop Gateways are also deployed into the public subnets for remote administration. Other components such as reverse proxy servers can be placed into these public subnets, if needed.

This VPC architecture uses two Availability Zones, each with its own distinct public and private subnets. We recommend that you leave plenty of unallocated address space to support the growth of your environment over time and to reduce the complexity of your VPC subnet design. Launch Wizard uses a default VPC configuration that provides plenty of address space by using the minimum number of private and public subnets. By default, Launch Wizard uses the following CIDR ranges.

<table>
<thead>
<tr>
<th>VPC</th>
<th>10.0.0.0/16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private subnets A</td>
<td>10.0.0.0/20</td>
</tr>
<tr>
<td></td>
<td>10.0.0.0/20</td>
</tr>
<tr>
<td></td>
<td>10.0.16.0/20</td>
</tr>
</tbody>
</table>
In addition, Launch Wizard provisions spare capacity for additional subnets to support your environment as it grows or changes over time. If you have sensitive workloads that must be completely isolated from the internet, you can create new VPC subnets using these optional address spaces.

### Security groups

Amazon EC2 instances must be associated with a security group, which acts as a stateful firewall. You control the network traffic entering or leaving the security group, and you can create rules that are defined by protocol, port number, and source/destination IP address, or other security groups. By default, all egress traffic from a security group is permitted. However, ingress traffic must be configured to allow the desired traffic to reach your instances.

The [Securing the Microsoft Platform on Amazon Web Services whitepaper](https://aws.amazon.com/about-aws/whats-new/wp-securing-microsoft-platform-amazon-web-services/) explains the different methods for securing your AWS infrastructure. Recommendations include providing isolation between application tiers by using security groups. We recommend that you tightly control ingress traffic in order to reduce the attack surface of your Amazon EC2 instances.

If you are deploying and managing your own AD DS installation, domain controllers and member servers will require several security group rules to allow traffic for services. These rules include AD DS replication, user authentication, Windows Time services, and Distributed File System (DFS). You should also consider restricting these rules to specific IP subnets that are used within your VPC.

For a detailed list of port mappings used by AWS CloudFormation, see the [Security best practices](p. 21) in this guide.


### Remote Desktop Gateway

When you design your architecture for highly available AD DS, you should also design for highly available and secure remote access. Launch Wizard optionally allows for deployment of a Remote Desktop (RD) Gateway server to manage your AD DS instances.

RD Gateway uses the Remote Desktop Protocol (RDP) over HTTPS to establish a secure, encrypted connection between remote administrators on the internet and Windows-based Amazon EC2 instances, without the need for a virtual private network (VPN) connection. This configuration reduces the attack surface of your Windows-based Amazon EC2 instances, while providing a remote administration solution for administrators.

**Important**

Never open up RDP to the entire internet even temporarily or for testing purposes. Always restrict ports and source traffic to the minimum necessary to support the functionality of the application.

### Active Directory

This section provides information about key design considerations specific to a Launch Wizard deployment of Active Directory Domain Services (AD DS) domain controllers on AWS.
Active Directory deployment topics

- Highly available directory domain services (p. 8)
- Active Directory DNS and DHCP inside the VPC (p. 8)
- DNS settings on Windows Servers instances (p. 9)
- Active Directory Certificate Services (p. 10)

Highly available directory domain services

Launch Wizard deploys two domain controllers in your AWS environment in two Availability Zones. This design provides fault tolerance and prevents a single domain controller failure from affecting the availability of the AD DS.

To strengthen the high availability of your architecture and help mitigate the impact of a possible disaster, each domain controller deployed by Launch Wizard is a global catalog server and an Active Directory DNS server.

When you choose to deploy self-managed domain controllers to the AWS Cloud, Launch Wizard automatically builds out an Active Directory Sites and Services configuration that supports a highly available AD DS architecture.

For information about creating sites, adding global catalog servers, and creating and managing site links, see the Microsoft Active Directory Sites and Services documentation.

Active Directory DNS and DHCP inside the VPC

Dynamic Host Configuration Protocol (DHCP) services are provided by default for your instances within a VPC. DHCP scopes do not need to be managed; they are created for the VPC subnets you define when you deploy your solution. These DHCP services cannot be disabled, so you must use them rather than deploying your own DHCP server.

The VPC also provides an internal DNS server. This DNS provides instances with basic name resolution services for internet access and is crucial for access to AWS service endpoints, such as AWS CloudFormation and Amazon S3 during bootstrapping.

Amazon-provided DNS server settings will be assigned to instances launched into the VPC based on a DHCP options set. DHCP options sets are used within an Amazon VPC to define scope options, such as the domain name or the name servers that should be handed to your instances via DHCP. Amazon-provided DNS is used only for public DNS resolution.

Because Amazon-provided DNS cannot be used to provide name resolution services for Active Directory, you must ensure that domain-joined Windows instances are configured to use Active Directory DNS.

Launch Wizard statically assigns Active Directory DNS server addresses on Windows instances. You can alternatively specify them using a custom DHCP options set. This allows you to assign your Active Directory DNS suffix and DNS server IP addresses as the name servers within the VPC through DHCP.

Note

The IP addresses in the domain-name-servers field are always returned in the same order. If the first DNS server in the list fails, instances should fall back to the second IP and continue to resolve host names successfully. However, during normal operations, the first DNS server listed will always handle DNS requests. If you want to ensure that DNS queries are distributed evenly across multiple servers, you should consider statically configuring DNS server settings on your instances.

For more information about creating a custom DHCP options set and associating it with your VPC, see Working with DHCP Options Sets in the Amazon VPC User Guide.
**Note**
If you choose to deploy self-managed domain controllers in the AWS Cloud, Launch Wizard adds the DNS suffix for your domain to the DNS suffixes list. The DNS settings on the local server point to the IP address of the first domain controller for all of the domain controllers in the infrastructure.

**DNS settings on Windows Servers instances**

To ensure that domain-joined Windows instances automatically register host (A) and reverse lookup (PTR) records with Active Directory-integrated DNS, set the properties of the network connection as shown in the following image.

The default configuration for a network connection is set to automatically register the connections address in DNS. In other words, the **Register this connection's addresses in DNS** option is selected for you automatically. This takes care of host (A) record dynamic registration. However, if you do not also select the second option, **Use this connection's DNS suffix in DNS registration**, dynamic registration of PTR records will not occur.

If you have a small number of instances in the VPC, you may choose to manually configure the network connection. For larger fleets, you can push this setting out to all of your Windows instances by using...
Active Directory Group Policy. For instructions about how to do this, see IPv4 and IPv6 Advanced DNS Tab in the Microsoft TechNet Library.

**Active Directory Certificate Services**

Launch Wizard sets up and deploys Active Directory Certificate Services (AD CS) with a new Active Directory infrastructure to issue and manage digital certificates in systems that use public key technologies. For more information about AD CS, see the Microsoft documentation.

**Self-managed domain controller architecture**

The Launch Wizard self-managed domain controller deployment sets up the following architecture.

- Domain controllers are deployed into two private VPC subnets in separate Availability Zones, which makes AD DS highly available.
- NAT gateways are deployed to public subnets, providing outbound internet access for instances in private subnets.
- Remote Desktop gateways are deployed in an Auto Scaling group in one Availability Zone to allow access to the domain controllers.

The specified version of Windows Server is used for the Remote Desktop Gateway instances and the domain controller instances. Launch Wizard deploys AWS resources, including a Systems Manager Automation document. When the second node is deployed, it initiates running the Automation document through Amazon EC2 user data. The automation workflow deploys the required components, finalizes the configuration to create a new AD forest, and promotes instances in two Availability Zones to Active Directory domain controllers.

To view architectural diagrams showing best practices for setting up an AD DS environment, see Active Directory Domain Services on AWS.

**Domain controller launch limits**

You can launch up to three domain controllers from a single Launch Wizard deployment per each AWS Region. If you want to add more domain controllers, you can launch additional stacks and add them to the same Active Directory Domain Services infrastructure.

**AWS Regions**

Launch Wizard uses AWS Secrets Manager during the provisioning of self-managed domain controllers into an environment. These services are not supported in all AWS Regions. For a current list of supported Regions, see the Service endpoints and quotas.

**Get started with AWS Launch Wizard for Active Directory**

This section contains information to set up your environment for Launch Wizard to deploy domain controllers.

**Topics include:**
- Access AWS Launch Wizard (p. 11)
- Set up for AWS Launch Wizard for Active Directory (p. 11)
- Deploy an application with AWS Launch Wizard for Active Directory (p. 15)
- Configure forest trust relationships (p. 20)
Access AWS Launch Wizard

You can launch AWS Launch Wizard from the AWS Launch Wizard console located at https://console.aws.amazon.com/launchwizard.

Set up for AWS Launch Wizard for Active Directory

Verify the following prerequisites to deploy self-managed domain controllers with AWS Launch Wizard.

Contents
- Active Directory (p. 11)
- AWS Identity and Access Management (IAM) (p. 12)
- Requirements for using custom AMIs (p. 13)
- Configuration settings (p. 14)

Active Directory

This section contains information to set up for deployment of domain controllers into an existing VPC and for deployment into an existing, on-premises Active Directory. It also contains information for setting up forest trusts.

Active Directory on EC2

If you deploy domain controllers into an existing VPC with an existing Active Directory, Launch Wizard requires domain administrator credentials to be added to Secrets Manager to join your domain controllers to Active Directory and promote them. In addition, a resource policy must be attached to the secret so that Launch Wizard can access the secret. Launch Wizard guides you through the following process of attaching the required policy to your secret.

How to attach the resource policy to your secret so that Launch Wizard can access the secret

1. Navigate to the Secrets Manager console.
2. Under **Secret name**, choose the name of your secret.
3. Under **Resource permissions**, choose **Edit permissions**.
4. Copy the following policy and paste it into the **Resource permissions** box, entering your account ID in the path of the IAM ARN.

```json
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Principal" : {
        "AWS" :
        "arn:aws:iam::<account-id>:role/service-role/AmazonEC2RoleForLaunchWizard"
      },
      "Resource" : "*"
    }
  ]
}
```
5. Choose **Save**.
6. Resume your Launch Wizard deployment.

The following key operations are performed against your Active Directory by Launch Wizard. These operations result in the creation of new records or entries in Active Directory.

- Creates a new Amazon EC2 instance and joins it to the domain.
- Creates ingress and egress rules to communicate with your domain controllers.
- Promotes the server to a domain controller in your domain.
- Updates local DNS on the new domain controllers to point to your DNS server.

On-premises Active Directory through AWS Direct Connect

If you are deploying domain controllers into an existing VPC and connecting to an on-premises Active Directory, ensure that the following prerequisites are in place.

- Make sure that you have connectivity between your AWS account and your on-premises network. You can establish a dedicated network connection from your on-premises network to your AWS account with AWS Direct Connect. For more information, see the AWS Direct Connect documentation.
- The domain functional level of your Active Directory domain controller must be Windows Server 2012 or later.
- The IP addresses of your DNS server must be either in the same VPC CIDR range as the one in which your Launch Wizard domain controllers will be created, or in the private IP address range.
- The firewall on the Active Directory domain controllers should allow the connections from the Amazon VPC from which you will create the Launch Wizard deployment. At a minimum, your configuration should include the ports mentioned in How to configure a firewall for Active Directory domains and trusts.

You can optionally perform the following step.

- Establish DNS resolution across your environments. For options on how to set this up, see How to Set Up DNS Resolution Between On-Premises Networks and AWS using AWS Directory Service and Amazon Route 53 or How to Set Up DNS Resolution Between On-Premises Networks and AWS Using AWS Directory Service and Microsoft Active Directory.

Trust relationships

If you are creating a forest trust relationship, you must complete the prerequisites for AWS Managed Active Directory (p. 11) before you set up the trust. For more information about creating forest trust relationships, see Configure forest trust relationships (p. 20).

AWS Identity and Access Management (IAM)

The following steps establish the required AWS Identity and Access Management (IAM) role and set up the IAM user for permissions.

One-time creation of IAM Role

On the Choose Application page of Launch Wizard, under Permissions, Launch Wizard displays the IAM role required for the Amazon EC2 instances that have been created by Launch Wizard to access other AWS services on your behalf. When you select Next, Launch Wizard attempts to discover the IAM role in your account. If the role exists, the role is attached to the instance profile for the Amazon EC2 instances that Launch Wizard will launch into your account. If the role does not exist, Launch Wizard attempts to create the role with the same name, AmazonEC2RoleForLaunchWizard. This role is comprised of two IAM managed policies: AmazonSSMManagedInstanceCore and
AmazonEC2RolePolicyForLaunchWizard. After the role is created, the IAM Administrator can delegate the application deployment process to another IAM user who, in turn, must have the Launch Wizard IAM managed policy described in the following section.

**IAM user setup**

To deploy self-managed domain controllers with Launch Wizard, you must create an Identity and Access Management (IAM) policy and attach it to your IAM user identity. The IAM policy defines the user permissions. If you do not already have an IAM user in your account, follow the steps listed in Create an IAM User in Your AWS Account.

When you have an IAM user in your account, create an IAM policy.

2. Choose Users from the left navigation pane.
3. Select the User name of the user to which you want to attach the policy.
4. Select Add permissions.
5. Select Attach existing policies directly.
6. Search for the policy named AmazonLaunchWizard_Fullaccess, and select the check box to the left of the policy name.
7. Select Next: Review.
8. Verify that the correct policy is listed, and then select Add permissions.

**Important**

Make sure that you log in with the user associated with the policy described in this procedure when you use Launch Wizard.

**Requirements for using custom AMIs**

We recommend that you use license-included Windows AMIs whenever possible. There are scenarios for which you may want to use a custom Windows AMI. For example, you may have existing licenses (BYOL), or you may have made changes to one of our public images and re-imaged it.

If you use license-included Windows AMIs, you are not required to perform any pre-checks on the AMI to ensure that it meets Launch Wizard requirements.

Launch Wizard relies on user data to begin the process of configuring domain controller instances that the service launches in your account. For more information, see User Data Scripts. By default, all Windows AMIs have user data execution enabled for the initial launch. To ensure that your custom AMIs are set up to run the User Data script at launch, follow the AWS-recommended method to prepare your AMIs using EC2Launch v2. For more information about how to prepare your custom AMI using the options to Shutdown with Sysprep or Shutdown without Sysprep, see Create a Standard Amazon Machine Image Using Sysprep or EC2Launch v2 and Sysprep. If you want to directly enable user data as part of the custom AMI creation process, follow the steps for Subsequent Reboots or Starts under Running Commands on Your Windows Instance at Launch.

If you use a custom AMI, the volume drive letter for the root partition should be C:, because EC2Launch v2 and EC2Config rely on this configuration to install the components.

While not exhaustive, the following requirements cover most of the configurations whose alteration might impact the successful deployment of a domain controllers using Launch Wizard.

<table>
<thead>
<tr>
<th>Windows Server 2016</th>
<th>Windows Server 2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>
Operating system requirements

- Windows Server 2016
- Windows Server 2019
- English language pack only
- The root volume drive for the custom AMI should be C:

AWS software and drivers

- AWS CloudFormation cfn-init script (See Bootstrapping AWS CloudFormation Windows stacks)
- EC2Launch (Windows Server 2016)
- AWS SSM (SSM agent must be installed)
- AWS Tools for Windows PowerShell
- Network drivers (SRIOV, ENA)
- Storage drivers (NVMe, AWS PV)

Configuration settings

The following configuration settings are applied when deploying self-managed domain controllers with Launch Wizard.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Applies to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current EC2Launch and SSM Agent</td>
<td>Windows Server 2016 and Windows Server 2019</td>
</tr>
<tr>
<td>Current AWS PV, ENA, and NVMe drivers</td>
<td>Windows Server 2016 and Windows Server 2019</td>
</tr>
<tr>
<td>Current SRIOV drivers</td>
<td>Windows Server 2016 and Windows Server 2019</td>
</tr>
<tr>
<td>Allow ICMP traffic through the firewall</td>
<td>Windows Server 2016 and Windows Server 2019</td>
</tr>
<tr>
<td>Allow RDP traffic through host firewall</td>
<td>Windows Server 2016 and Windows Server 2019</td>
</tr>
<tr>
<td>RealTimeIsUniversal registry key set</td>
<td>Windows Server 2016 and Windows Server 2019</td>
</tr>
</tbody>
</table>

The following AMI settings can impact the Launch Wizard deployment:

System Time

RealTimeIsUniversal. If disabled, Windows system time drifts when the time zone is set to a value other than UTC.

Windows Firewall

In most cases, Launch Wizard configures the correct protocols and ports. However, custom Windows Firewall rules could impact the cluster service. To ensure that your custom AMI works with Launch Wizard, see Service overview and network port requirements for Windows.
Remote Desktop

Service Start. Remote Desktop service must be enabled.

Remote Desktop Connections. Must be enabled.

Network Interface

DHCP Service Startup. DHCP service should be enabled.

DHCP on Ethernet. DHCP should be enabled.

PowerShell

Execution Policy. The execution policy in all AWS license-included AMIs is set to Unrestricted. We recommend that you set this policy to Unrestricted when you set up domain controllers using AWS Launch Wizard You can change the policy when setup is complete.

Deploy an application with AWS Launch Wizard for Active Directory

The following steps guide you through a domain controller deployment with AWS Launch Wizard after you have launched it from the console.

1. When you select Create deployment from the AWS Launch Wizard landing page, you are directed to the Choose application page where you are prompted to select the type of application that you want to deploy. Select Active Directory and choose Create deployment.

2. After you select the application to configure for deployment, under Permissions, Launch Wizard displays the AWS Identity and Access Management (IAM) role required for Launch Wizard to access other AWS services on your behalf. For more information about setting up the IAM role for Launch Wizard, see AWS Identity and Access Management (IAM) (p. 12). Choose Next.

3. On the Configure application settings page, enter specifications for the new deployment. The following tabs provide information about the specification fields.

General settings and Active Directory (AD) installation

General settings

- Deployment name. Enter a unique application name for your deployment.
- Simple Notification Service (SNS) topic ARN (Optional). Specify an Amazon SNS topic where AWS Launch Wizard can send notifications and alerts. For more information, see the Amazon Simple Notification Service Developer Guide.
- Enable rollback on failed deployment. By default, if a deployment fails, your provisioned resources will not be rolled back/deleted. This default configuration helps you to troubleshoot errors at the resource level as you debug deployment issues. If you want your provisioned resources to be immediately deleted if a deployment fails, select the check box.

Installation type

Choose whether you want to deploy Active Directory on Amazon EC2 instances or on premises. When you deploy Active Directory on Amazon EC2, it is deployed in a VPC configured for high availability.

For the Domain settings that follow, choose the tab that applies to the AMI type you plan to use for the deployment: Domain settings — license-included AMI or Domain settings — custom AMI.
Domain settings – license-included AMI

Enter the following specifications to configure your domain controllers on a license-include AMI.

**Number of domain controllers**

Select the number of servers on which you want to install Active Directory.

**Domain controller details — optional**

For each domain controller, enter a name.

**AMI installation type**

Choose to use an AWS license-included AMI with Windows installed.

- **License-included AMI.** Choose the license-included AMI you want to use to launch your domain controller environment.
- **Active Directory (AD) settings.** Choose whether you want to add domain controllers to an existing Active Directory or you want to create a new Active Directory.

**Add domain controllers to an existing Active Directory**

- **Active Directory DNS domain name.** Enter the DNS domain name for your directory to connect to an existing Active Directory. For more information about DNS support of Active Directory, see the [Microsoft Active Directory documentation](#).
- **Domain NetBIOS name.** The NetBIOS name is the subdomain of the DNS domain name. Enter the NetBIOS name to connect to Active Directory.
- **Domain Administrator secret name.** Enter the name of the AWS Secrets Manager name of the Domain Administrator to access stored credentials.
- **Add permissions to AWS Secrets Manager — required.** Follow the instructions to add the required resource policy for Launch Wizard to access Secrets Manager.
- **Domain DNS IP address for resolution.** Enter the IP address of the DNS server to which you are connecting.

**Create a new Active Directory**

- **Active Directory DNS domain name.** Enter a DNS domain name for your new Active Directory. For more information about DNS support of Active Directory, see the [Microsoft Active Directory documentation](#).
- **Domain NetBIOS name.** The NetBIOS name is the sub-domain of the DNS domain name. Enter a NetBIOS name to connect to Active Directory.
- **Domain Administrator username.** Launch Wizard applies the Administrator username by default.
- **Additional settings — optional.**
  - **AD Certificate Services (AD CS) installation.** Select whether you want to install Active Directory Certificate Services. For more information about AD CS, see the [Microsoft documentation](#).
  - **ADCS Certificate Server name (if selected).** If you selected to install AD CS, enter the AD CS certificate server name to use.
  - **Key length.** Select the key length of the encryption key.
  - **Key hash algorithm.** Select the algorithm used for key encryption.
  - **Key validity.** Select a key validity period in years. The key validity period determines the expiration date of the certificates.
• **Forest trust.** Select whether you want to create a forest trust with another forest. A forest trust is a transitive trust between two forests that allows users in any domain in one forest to be authenticated by any domain in another forest. For more information, see the Microsoft Active Directory documentation.

• **Fully qualified domain name of forest.** If you selected to create a forest trust, enter the fully qualified domain name of the forest.

• **Direction of trust.** Select the direction of the trust. Trusts can be one-way or two-way.

• **Conditional forwarder IP address.** Enter the IP address of the conditional forwarder that is set up in your forest.

**Domain settings – custom AMI**

Enter the following specifications to configure your domain controllers on a custom AMI.

**Number of domain controllers**

Select the number of servers on which you want to install Active Directory.

**Domain controller details — optional**

For each domain controller, enter a name.

**AMI installation type**

Choose to use a custom AMI. Verify that the AMI is configured to meet all of the required installation parameters described in Requirements for using custom AMIs (p. 13).

• **Custom AMI ID.** Select the custom AMI you want to use to launch your domain controller environment.

• **Active Directory DNS domain name.** Enter the DNS domain name for your directory to connect to an existing Active Directory. For more information about DNS support of Active Directory, see the Microsoft Active Directory documentation.

• **Domain NetBIOS name.** The NetBIOS name is the subdomain of the DNS domain name. Enter the NetBIOS name to connect to Active Directory.

• **Domain Administrator secret name.** Enter the name of the AWS Secrets Manager name of the Domain Administrator to access stored credentials.

• **Add permissions to AWS Secrets Manager — required.** Follow the instructions to add the required resource policy for Launch Wizard to access AWS Secrets Manager.

• **Domain DNS IP address for resolution.** Enter the IP address of the DNS server to which you are connecting.

**Connectivity**

Enter specifications for how you want to connect to your application instance and what kind of virtual private cloud (VPC) you want to set up.

**Key pair name**

• Select an existing key pair from the dropdown list or create a new one. If you select **Create new key pair name** to create a new key pair, you are directed to the Amazon EC2 console. From there, under **Network and Security**, choose **Key Pairs**. Choose **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**.
Important
This is the only chance for you to save the private key file, so be sure to download it and save it in a safe place. You must provide the name of your key pair when you launch an instance, and provide the corresponding private key each time that you connect to the instance.

Return to the Launch Wizard console and choose the refresh button next to the **Key Pairs** dropdown list. The newly created key pair appears in the dropdown list. For more information about key pairs, see [Amazon EC2 Key Pairs and Windows Instances](https://docs.aws.amazon.com/AmazonEC2/latest/UserGuide/key-pairs.html).

**Virtual Private Cloud (VPC).** Choose whether you want to use an existing VPC or create a new VPC.

**Select Virtual Private Cloud (VPC)**

- **Select Virtual Private Cloud (VPC)** option. Choose the VPC that you want to use from the dropdown list. Your VPC must contain one public subnet and, at least, two private subnets. Your VPC must be associated with a **DHCP Options Set** to enable DNS translations to work. The private subnets must have outbound connectivity to the internet and other AWS services (Amazon S3, AWS CloudFormation, SSM, CloudWatch Logs). We recommend that you enable this connectivity with a NAT Gateway. For more information about NAT Gateways, see [NAT Gateways](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-dns.html) in the Amazon VPC User Guide.

- **For deployment of a new Active Directory on Amazon EC2**, choose your **Remote Desktop Gateway (RDG) preferences**. This option is not available when you add domain controllers to an existing infrastructure.

  - Choose whether you want to set up Remote Desktop Gateway now, or set it up later.

  - If you choose to set up RDG now, enter the **Remote Desktop Gateway CIDR**. Select **Custom IP** from the dropdown list and enter the CIDR block. If you do not specify any value for the Custom IP parameter, Launch Wizard does not set the inbound RDP access (Port 3389) from any IP. You can choose to do this later by modifying the security group settings by using the Amazon EC2 console. See [Adding a Rule for Inbound RDP Traffic to a Windows Instance](https://docs.aws.amazon.com/AmazonEC2/latest/UserGuide/using-rdp.html) for instructions on adding a rule that allows inbound RDP traffic to your RDGW instance.

- **Availability Zone configuration.** You must choose at least two Availability Zones, with one private subnet for each zone that you select. From the dropdown lists, select the **Availability Zones** within which you want to deploy the domain controllers. Depending on the number of domain controllers you plan to deploy, you may have to specify a **private subnet** for each of them. Cross-Region replication is not supported.

**To create a private subnet**

If a subnet does't have a route to an internet gateway, the subnet is known as a private subnet. To create a private subnet, you can use the following steps. We recommend that you enable the outbound connectivity for each of your selected private subnets using a NAT Gateway. To enable outbound connectivity from private subnets with public subnet, see the steps in [Creating a NAT Gateway](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html) to create a NAT Gateway in your chosen public subnet. Then, follow the steps in [Updating Your Route Table](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/using-routes.html) for each of your chosen private subnets.

- Follow the steps in [Creating a Subnet](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/creating-subnet.html) in the Amazon VPC User Guide using the existing VPC you will use in AWS Launch Wizard.

- When you create a VPC, it includes a main route table by default. On the **Route Tables** page in the Amazon VPC console, you can view the main route table for a VPC by looking for **Yes** in the **Main** column. The main route table controls the routing for all subnets that are not explicitly associated with any other route table. If the main route table for your VPC has an outbound route to an internet gateway, then any subnet created using the
previous step, by default, becomes a public subnet. To ensure the subnets are private, you may need to create separate route tables for your private subnets. These route tables must not contain any routes to an internet gateway. Alternatively, you can create a custom route table for your public subnet and remove the internet gateway entry from the main route table.

- **Outbound connectivity.** Confirm that you have set up a public subnet and that each of the private subnets are configured to send traffic.

### Create a new virtual private cloud (VPC)

AWS Launch Wizard creates your VPC. You can optionally enter a VPC name tag.

- **Remote Desktop Gateway CIDR.** Select Custom IP from the dropdown list. Enter the CIDR block. If you do not specify any value for the Custom IP parameter, Launch Wizard does not set the inbound RDP access (Port 3389) from any IP. You can choose to do this later by modifying the security group settings via the Amazon EC2 Console. See Adding a Rule for Inbound RDP Traffic to a Windows Instance for instructions on adding a rule that allows inbound RDP traffic to your RDGW instance.

4. When you are satisfied with your configuration selections, select **Next.** If you don't want to complete the configuration, select **Cancel.** When you select **Cancel,** all of the selections on the settings page are lost and you are returned to the landing page. To go to the previous screen, select **Previous.**

5. After configuring your application, you are prompted to define the infrastructure requirements for the new deployment on the **Define infrastructure requirements** page.

### Storage and Compute

Enter your instance size preferences and view recommended resources.

- **Instance sizing** Choose whether you want to base the size of your deployed instances on infrastructure requirements or you want to select the instance type. If you choose to select the infrastructure requirements, enter the number of Active Directory users in the environment. If you choose to select the instance type, select the instance type from the dropdown.

- **Recommended resources.** Launch Wizard displays the system-recommended resources based on your infrastructure selections. If you want to change the recommended resources, select different infrastructure requirements.

6. When you are satisfied with your infrastructure selections, choose **Next.** If you don't want to complete the configuration, select **Cancel.** When you select **Cancel,** all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select **Previous.**

7. On the **Review** page, review your configuration details. If you want to make changes, select **Edit** or **Previous.** To stop, select **Cancel.** When you select **Cancel,** all of the selections on the specification page are lost and you are returned to the landing page. If you are ready to deploy, read and select the check box next to the **Acknowledgment.** Then choose **Deploy.**

8. Launch Wizard validates the inputs and notifies you if something must be addressed.

9. When validation is complete, Launch Wizard deploys your AWS resources and configures your domain controllers. Launch Wizard provides you with status updates about the progress of the deployment on the **Deployments** page. From the **Deployments** page, you can also view the list of current and previous deployments.

10. When your deployment is ready, you see notification that your domain controllers are successfully deployed. If you have set up SNS notification, you are also alerted through Amazon SNS. You can manage and access all of the resources related to your domain controllers by selecting **Manage.**

11. When the domain controllers are deployed, you can access your Amazon EC2 instances through the Amazon EC2 console. You can also use **AWS SSM** to manage your domain controllers for future updates and patches through built-in integration through resource groups.
Configure forest trust relationships

For existing VPCs, you can optionally set up forest trusts with other Active Directory forests. For the required prerequisites to set up forest trusts, see Trust relationships (p. 12) in the Prerequisites section of this guide.

For more information about when to create a trust relationship with other Active Directory forests, see When to Create a Trust Relationship in the AWS Directory Service Administration Guide.

For information about how to add IP routes when the DNS servers for the networks of the other directories use public IP addresses, see Adding IP Routes When Using Public IP Addresses in the AWS Directory Service Administration Guide.

For steps to create a trust relationship, see Tutorial: Create a Trust Relationship Between Your AWS Managed Microsoft AD and Your On-Premises Domain in the AWS Directory Service Administration Guide.

Manage application resources with AWS Launch Wizard for Active Directory

After you deploy your self-managed domain controllers, you can manage them by following these steps.

1. From the navigation pane, choose Deployments.
2. From the Deployments page, select Actions. You can select to do the following:
   1. Manage resources on the EC2 console. You are taken to the Amazon EC2 console, where you can view and manage your domain controller resources. For example, you can view and manage EC2, Amazon EBS, Active Directory, VPC, subnets, NAT Gateways, and Elastic IPs.
   2. View resource group with SSM. You are taken to the Systems Manager console to view your resource groups.
   3. View CloudWatch application logs. You are taken to CloudWatch Logs, where you can monitor, store, and access your SQL Server Always On application log files.
   4. View your CloudFormation template. This is the CloudFormation template created by your most recent deployment, and it can be accessed through the CloudFormation console. For help with finding and using your CloudFormation template, see Viewing AWS CloudFormation Stack Data and Resources on the AWS Management Console.
3. To delete a deployment, select the application that you want to delete and select Delete. You are prompted to confirm your action.
   Important
   You lose all specification settings for the domain controllers when you delete a deployment. AWS Launch Wizard attempts to delete only the AWS resources that it created in your account as part of the deployment. If you created resources outside of Launch Wizard, for example resources that reside in a VPC created by Launch Wizard, the deletion may fail. Launch Wizard does not delete any Active Directory objects in your Active Directory, nor any of the records in your DNS server. Launch Wizard has no control over your Active Directory domain user password over time, which is required to clean up Active Directory objects or DNS records. We recommend that you remove these entries from your Active Directory after Launch Wizard deletes the deployment. For key operations performed against your Active Directory resulting in new records or entries, see Active Directory on EC2 (p. 11).
4. To further investigate details regarding your domain controller resources, select the Application name. You can then view the Deployment events and Summary details for your application by using the tabs at the top of the page.
High availability and security best practices for AWS Launch Wizard for Active Directory

The domain controller architecture created by AWS Launch Wizard supports AWS best practices for high availability and security as promoted by the AWS Well-Architected Framework.

Topics
- High availability (p. 21)
- Security in Launch Wizard for Active Directory (p. 21)

High availability

With Amazon EC2, you can set the location of instances in multiple locations composed of AWS Regions and Availability Zones. Regions are dispersed and located in separate geographic areas. Availability Zones are distinct locations within a Region that are engineered to be isolated from failures in other Availability Zones. Availability Zones provide inexpensive, low-latency network connectivity to other Availability Zones in the same Region.

When you launch your instances in different Regions, you can set your domain controllers to be closer to specific customers, or to meet legal or other requirements. When you launch your instances in different Availability Zones, you can protect your domain controllers from the failure of a single location.

Security in Launch Wizard for Active Directory

Launch Wizard creates a number of security groups and rules for you. When Amazon EC2 instances are launched, they must be associated with a security group, which acts as a stateful firewall. You have complete control over the network traffic entering or leaving the security group. You can also build granular rules that are scoped by protocol, port number, and source or destination IP address or subnet. By default, all outbound traffic from a security group is permitted. Inbound traffic, on the other hand, must be configured to allow the appropriate traffic to reach your instances.

The Securing the Microsoft Platform on Amazon Web Services whitepaper discusses the different methods for securing your AWS infrastructure. Recommendations include providing isolation between application tiers using security groups. We recommend that you tightly control inbound traffic to reduce the attack surface of your EC2 instances.

Launch Wizard configures the necessary security groups for you, which are listed in the following table.

<table>
<thead>
<tr>
<th>Security group</th>
<th>Associated with</th>
<th>Inbound source</th>
<th>Port(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DomainControllerSG</td>
<td>DC1, DC2, DC3, CA</td>
<td>VPCCIDR</td>
<td>UDP53, TCP3389, TCP445, All ICMP-IPV4, IpProtocol-1, FromPort-1, ToPort-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DomainControllerSG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DomainMemberTCPKG</td>
<td>TCP49152-65535, TCP445, ICMP-1, TCP135, TCP139, TCP3269, TCP464,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Troubleshoot AWS Launch Wizard for Active Directory

Each deployment in your account in the same AWS Region can be uniquely identified by the name specified at the time of a deployment. The deployment name can be used to view the details related to the deployment on the **Deployments** page of the Launch Wizard console.

This section describes steps to help you troubleshoot deploying domain controllers with Launch Wizard for Active Directory.

**Contents**

- Launch Wizard provisioning events (p. 23)
- CloudWatch Logs (p. 23)
- AWS CloudFormation stack (p. 23)
- Errors (p. 23)

<table>
<thead>
<tr>
<th>Security group</th>
<th>Associated with</th>
<th>Inbound source</th>
<th>Port(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DomainMemberUDP websocket</td>
<td></td>
<td>TCP5722, TCP389, TCP9389, TCP3268, TCP88, TCP636</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DomainMemberUDP websocket</td>
<td>UDP49152-65535, UDP53, UDP389, UDP445, UDP138, UDP5355, UDP123, UDP88</td>
</tr>
<tr>
<td>DomainMemberTCP socket</td>
<td>RDGW</td>
<td>TCP464, TCP5722, TCP 49152-65535, TCP 389, TCP 445, TCP 3389, TCP9389, TCP3268, TCP123, TCP5985, TCP88, TCP139, TCP135, TCP636, TCP3269, TCP53</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ADServer1PrivateIp, ADServer2PrivateIp, ADServer3PrivateIp</td>
<td>TCP3389</td>
</tr>
<tr>
<td>DomainMemberUDP socket</td>
<td>RDGW</td>
<td>UDP445, UDP138, UDP49152-65535, UDP464, UDP5355, UDP137, UDP53, UDP389, UDP88</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ADServer1PrivateIp, ADServer2PrivateIp, ADServer3PrivateIp</td>
<td>TCP3389</td>
</tr>
<tr>
<td>RDGWSecurityGroup</td>
<td>RDGW1, RDGW2</td>
<td>RDGW CIDR</td>
<td>TCP3389</td>
</tr>
</tbody>
</table>

**Important**

Always restrict ports and source traffic to the minimum necessary to support the functionality of the application.
Launch Wizard provisioning events

Launch Wizard captures events from AWS CloudFormation to track the status of an ongoing application deployment. If an application deployment fails, you can view the deployment events for this application by selecting Deployments from the navigation pane. A failed event shows a status of Failed along with a failure message.

CloudWatch Logs

Launch Wizard streams provisioning logs from all of the AWS log sources, such as AWS CloudFormation and PowerShell DSC scripts to CloudWatch Logs. You can view the CloudWatch Logs for a given application name on the CloudWatch console for the log group name LaunchWizard-APPLICATION_NAME and log stream ApplicationLaunchLog.

AWS CloudFormation stack

Launch Wizard uses AWS CloudFormation to provision the infrastructure resources of an application. CloudFormation stacks can be found in your account using the CloudFormation describe-stacks API. Launch Wizard launches various stacks in your account for validation and application resource creation. The following are the relevant filters for the describe-stacks API.

- Application Resources
  - LaunchWizard-APPLICATION_NAME. This stack includes all of the resource creation events for resources created by the deployment.
  - LaunchWizard-STACK_NAME-TEMPLATE_NAME. This log includes all of the logs from each PowerShell script run from within the instance.

You can view the status of these CloudFormation stacks. If any of them fail, you can view the cause of failure.

Errors

Failed to create Forest Trust

- **Cause:** Forest trust fails because of the lack of connectivity between the two domain controllers.
- **Solution:** Ensure connectivity between the VPCs and ensure the ports are open between them. See Configure forest trust relationships (p. 20) for more details about configuring forest trusts.

The requested instance type is not supported in the requested Availability Zone

- **Cause:** This failure can occur when you launch either your RDGW instance or your Active Directory Server instance, or during the validation of the instances that Launch Wizard launches in your selected subnets.
- **Solution:** Choose a different Availability Zone and retry the deployment from the initial page of the Launch Wizard console.

Instance stabilization error

- **Cause:** This failure can occur when the EC2 instance used for validation fails to stabilize. When this happens, the EC2 instance is unable to communicate to the CloudFormation service to signal completions, resulting in WaitCondition errors.
- **Solution:** Please contact AWS Support.
AWS Launch Wizard for Amazon Elastic Kubernetes Service

This section of the AWS Launch Wizard documentation provides guidance for deploying self-managed Amazon EKS resources.

Topics
- What is AWS Launch Wizard for Amazon Elastic Kubernetes Service? (p. 24)
- Get Started with AWS Launch Wizard for Amazon Elastic Kubernetes Service (p. 26)
- Test the deployment (p. 40)
- Best practices (p. 40)
- Troubleshoot AWS Launch Wizard for Amazon EKS (p. 41)

What is AWS Launch Wizard for Amazon Elastic Kubernetes Service?

AWS Launch Wizard for Amazon Elastic Kubernetes Service (Amazon EKS) guides you through the sizing, configuration, and deployment of an Amazon EKS control plane, connecting worker nodes to the cluster, and configuring a bastion host for cluster admin operations. Additionally, the deployment provides custom resources that enable you to deploy and manage your Kubernetes applications using AWS CloudFormation by declaring Kubernetes manifests or Helm charts directly in CloudFormation templates.

Contents
- Deployment options (p. 24)
- Components (p. 24)

Deployment options

Launch Wizard for Amazon EKS supports the following deployment types:
- Deploy an Amazon EKS cluster into a new Amazon Virtual Private Cloud in your AWS account.
- Deploy an Amazon EKS cluster into an existing Amazon Virtual Private Cloud in your AWS account.

Components

An Amazon EKS environment deployed with Launch Wizard will include the following components:
- A highly available architecture that spans three Availability Zones.
- In one public subnet, a Linux bastion host in an Auto Scaling group to allow inbound Secure Shell (SSH) access to Amazon Elastic Compute Cloud (Amazon EC2) instances in private subnets. The bastion host is also configured with the Kubernetes kubectl command line interface (CLI) for managing the Kubernetes cluster.
- An Amazon EKS cluster, which creates the Kubernetes control plane.
- In the private subnets, a group of Kubernetes nodes.
• Resource Groups that contain all the resources created with Launch Wizard.

Additionally, a new VPC deployment includes the following components:

• A VPC configured with public and private subnets according to AWS best practices, to provide you with your own virtual network in AWS.
• In the public subnets, managed NAT gateways to allow outbound internet access for resources in the private subnets.
Get Started with AWS Launch Wizard for Amazon Elastic Kubernetes Service

This section contains information to help you set up your environment to deploy Amazon EKS with Launch Wizard. When your environment is set up, you can deploy Amazon EKS application with Launch Wizard by following the steps and parameter specification details provided in this section.

Topics to help you get started:
- Access AWS Launch Wizard (p. 26)
- Set up (p. 26)
- Deploy Amazon Elastic Kubernetes Service into a new Amazon Virtual Private Cloud (p. 27)
- Deploy Amazon Elastic Kubernetes Service into an existing Amazon Virtual Private Cloud (p. 33)

Access AWS Launch Wizard

You can launch AWS Launch Wizard from the AWS Launch Wizard console located at https://console.aws.amazon.com/launchwizard.

Set up

Verify the following prerequisites to deploy an Amazon EKS application with AWS Launch Wizard.
- Specialized knowledge (p. 26)
- Amazon Web Services account (p. 26)
- Technical requirements (p. 26)
- Service Quotas (p. 27)
- IAM permissions (p. 27)

Specialized knowledge

This deployment requires a moderate level of familiarity with AWS services. If you're new to AWS, see Getting Started Resource Center and AWS Training and Certification. These sites provide materials for learning how to design, deploy, and operate your infrastructure and applications on the AWS Cloud.

This Launch Wizard assumes familiarity with Kubernetes concepts and usage.

Amazon Web Services account

If you don't already have an AWS account, create one at http://aws.amazon.com/ by following the on-screen instructions. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.

Your AWS account is automatically signed up for all AWS services. You are charged only for the services you use.

Technical requirements

Before you start the Launch Wizard deployment, review the following information and ensure that your account is properly configured. Otherwise, deployment might fail.
Service Quotas

If necessary, request service quota increases for the following resources. You might need to request increases if your existing deployment currently uses these resources, and if this Launch Wizard deployment could result in exceeding the default quotas. The Service Quotas console displays your usage and quotas for some aspects of some services. For more information, see What is Service Quotas? and AWS service quotas.

Existing VPC Service Quotas:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default quota</th>
<th>This deployment uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic IP Addresses</td>
<td>5 per Region</td>
<td>1</td>
</tr>
<tr>
<td>VPC security groups</td>
<td>300 per account</td>
<td>3</td>
</tr>
<tr>
<td>IAM roles</td>
<td>1,000 per account</td>
<td>9</td>
</tr>
<tr>
<td>Auto Scaling groups</td>
<td>200 per Region</td>
<td>2</td>
</tr>
<tr>
<td>Amazon EC2 On-Demand Instances (Standard)</td>
<td>5 per Region</td>
<td>4</td>
</tr>
</tbody>
</table>

New VPC Service Quotas:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default quota</th>
<th>This deployment uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPCs</td>
<td>5 per Region</td>
<td>1</td>
</tr>
<tr>
<td>Elastic IP Addresses</td>
<td>5 per Region</td>
<td>4</td>
</tr>
<tr>
<td>Internet Gateway</td>
<td>5 per Region</td>
<td>1</td>
</tr>
<tr>
<td>VPC security groups</td>
<td>300 per account</td>
<td>3</td>
</tr>
<tr>
<td>IAM roles</td>
<td>1,000 per account</td>
<td>9</td>
</tr>
<tr>
<td>Auto Scaling groups</td>
<td>200 per Region</td>
<td>2</td>
</tr>
<tr>
<td>Amazon EC2 On-Demand Instances (Standard)</td>
<td>5 per Region</td>
<td>4</td>
</tr>
</tbody>
</table>

IAM permissions

Before deploying the Launch Wizard application, you must sign in to the AWS Management Console with IAM permissions for the resources that the templates deploy. The AdministratorAccess managed policy within IAM provides sufficient permissions, although your organization may choose to use a custom policy with more restrictions. For more information, see AWS managed policies for job functions.

Deploy Amazon Elastic Kubernetes Service into a new Amazon Virtual Private Cloud

The following steps guide you through an Amazon EKS deployment with AWS Launch Wizard after you have launched it from the console.
1. When you select **Choose application** from the AWS Launch Wizard landing page, you are directed to the Choose application wizard where you are prompted to select the type of application that you want to deploy.

2. Select **Amazon EKS**, select **Deploy Amazon EKS into a new VPC**, then select **Create deployment**.

3. You are prompted to enter the specifications for the new deployment. The following tabs provide information about the specification fields of the deployment model.

**General**

- **Deployment name.** Enter a unique application name for your deployment.

- **Amazon Simple Notification Service (SNS) topic ARN — optional.** Specify an Amazon SNS topic where AWS Launch Wizard can send notifications and alerts. For more information, see the Amazon Simple Notification Service Developer Guide.

- **Deactivate rollback on failed deployment.** By default, if a deployment fails, your provisioned resources will be deleted. You can enable this setting during deployment to prevent this behavior.

- **Tags - optional.** Enter a key and value to assign metadata to your deployment. For help with tagging, see Tagging Your Amazon EC2 Resources.

**Network configuration**

- **Key pair name.** Select an existing key pair from the dropdown list or create a new one. If you select **Create new key pair name**, you are directed to the Amazon EC2 console. From there, under **Network and Security**, choose **Key Pairs**. Choose **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**.

  **Important**
  This is the only opportunity for you to save the private key file. Download it and save it in a safe place. You must provide the name of your key pair when you launch an instance and provide the corresponding private key each time that you connect to the instance. Return to the Launch Wizard console and choose the refresh button next to the **Key Pairs** dropdown list. The newly created key pair appears in the dropdown list. For more information about key pairs and Linux instances, see Amazon EC2 Key Pairs and Linux Instances. For more information about key pairs and Windows instances, see Amazon EC2 Key Pairs and Windows Instances.

- **Allowed external access CIDR: Allowed CIDR block for external access to the deployed instances.**

- **VPC settings:** Launch Wizard creates your VPC in this case. Input fields that define the VPC configuration are shown in the following list.

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Availability Zones (NumberOfAZs)</td>
<td>3</td>
<td>Number of Availability Zones to use in the VPC. A minimum number of 2 and maximum number of 3 Zones is allowed. This must match the value entered for the AvailabilityZones parameter.</td>
</tr>
<tr>
<td>VPC CIDR (VPCCIDR)</td>
<td>10.0.0.0/16</td>
<td>CIDR block for the VPC.</td>
</tr>
</tbody>
</table>
### Deploy to a new VPC

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private subnet 1 CIDR</td>
<td>10.0.0.0/19</td>
<td>CIDR block for private subnet 1, located in Availability Zone 1.</td>
</tr>
<tr>
<td>(PrivateSubnet1CIDR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Private subnet 2 CIDR</td>
<td>10.0.32.0/19</td>
<td>CIDR block for private subnet 2, located in Availability Zone 2.</td>
</tr>
<tr>
<td>(PrivateSubnet2CIDR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Private subnet 3 CIDR</td>
<td>10.0.64.0/19</td>
<td>(Optional) CIDR block for private subnet 3, located in Availability Zone 3.</td>
</tr>
<tr>
<td>(PrivateSubnet3CIDR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Public subnet 1 CIDR</td>
<td>10.0.128.0/20</td>
<td>CIDR block for the public (DMZ) subnet 1, located in Availability Zone 1.</td>
</tr>
<tr>
<td>(PublicSubnet1CIDR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Public subnet 2 CIDR</td>
<td>10.0.144.0/20</td>
<td>CIDR block for the public (DMZ) subnet 2, located in Availability Zone 2.</td>
</tr>
<tr>
<td>(PublicSubnet2CIDR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Public subnet 3 CIDR</td>
<td>10.0.160.0/20</td>
<td>(Optional) CIDR block for the public (DMZ) subnet 3, located in Availability Zone 3.</td>
</tr>
<tr>
<td>(PublicSubnet3CIDR)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### EKS configuration

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config set name</td>
<td>Blank string</td>
<td>(Optional) This parameter is used to map advanced parameters to an EKS cluster. You can keep it blank unless you are using an advanced configuration stack. If you launched an advanced configuration stack and want to apply its values to this cluster, this name must match the ConfigSetsetName parameter for the stack. If kept blank, a new Config set is created using default values.</td>
</tr>
<tr>
<td>(ConfigSetName)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP proxy (HttpProxy)</td>
<td>Blank string</td>
<td>(Optional) HTTP(S) proxy configuration. If provided, all worker nodes and pod egress traffic use this proxy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example: <a href="http://10.101.0.100:3128/">http://10.101.0.100:3128/</a></td>
</tr>
<tr>
<td>Parameter label (name)</td>
<td>Default value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Per-account shared resources</td>
<td>AutoDetect</td>
<td>This EKS deployment creates several IAM roles and instance profiles that are intended to be deployed only once in an AWS account. If you already have an existing Launch Wizard EKS application deployed in this AWS account, in this AWS Region or another, you must choose No to skip creation of the per-account shared resources.</td>
</tr>
<tr>
<td>Per-Region shared resources</td>
<td>AutoDetect</td>
<td>This EKS deployment sets up several resources such as helper Lambda functions, an S3 bucket for staging assets, and AWS CloudFormation macros that are intended to be deployed once for each AWS Region and shared in future deployments of Launch Wizard EKS in that Region. If you already have an existing Launch Wizard EKS application deployed in this account in this Region, you must choose No to skip creation of the per-Region shared resources.</td>
</tr>
<tr>
<td>Provision bastion host</td>
<td>Activated</td>
<td>Skip creating a bastion host by deactivating this option.</td>
</tr>
<tr>
<td>EKS cluster name</td>
<td>Blank string</td>
<td>(Optional) Name for the EKS cluster. If kept blank, one is automatically generated. This must be unique within the Region.</td>
</tr>
<tr>
<td>EKS public access endpoint</td>
<td>Deactivated</td>
<td>Configure access to the Kubernetes API server endpoint from outside of your VPC.</td>
</tr>
<tr>
<td>Additional EKS admin ARN (IAM user)</td>
<td>Blank string</td>
<td>(Optional) IAM user ARN to be granted administrative access to the EKS cluster.</td>
</tr>
<tr>
<td>Additional EKS admin ARN (IAM role)</td>
<td>Blank string</td>
<td>(Optional) IAM role ARN to be granted administrative access to the EKS cluster.</td>
</tr>
<tr>
<td>Fargate namespaces</td>
<td>Blank string</td>
<td>(Optional) Comma-separated list of namespaces for which Fargate should be enabled.</td>
</tr>
</tbody>
</table>
EKS node group configuration

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes (NumberOfNodes)</td>
<td>3</td>
<td>Number of Amazon EKS node instances. The default is one for each of the three Availability Zones.</td>
</tr>
<tr>
<td>Maximum number of nodes (MaxNumberOfNodes)</td>
<td>3</td>
<td>Maximum number of Amazon EKS node instances. The default is three.</td>
</tr>
<tr>
<td>Node group OS (NodeGroupOS)</td>
<td>Amazon Linux 2</td>
<td>Operating system to use for node instances. Choose Bottlerocket for the Amazon purpose-built container OS (unmanaged node groups only). Note that if you choose Windows, an additional Amazon Linux node group is created.</td>
</tr>
<tr>
<td>Node group type (NodeGroupType)</td>
<td>Managed</td>
<td>Choose Unmanaged to create an Auto Scaling group without using the EKS-managed node groups feature.</td>
</tr>
<tr>
<td>Node instance family (NodeInstanceFamily)</td>
<td>Standard</td>
<td>Choose the instance family to match the value of Node instance type.</td>
</tr>
</tbody>
</table>

Kubernetes add-ins

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS load balancer controller (ALBIngressController)</td>
<td>Activated</td>
<td>You can deactivate deploying the AWS load balancer controller. If you activate deployment of the AWS load balancer controller, a Helm chart for this component is deployed.</td>
</tr>
<tr>
<td>Cluster autoscaler (ClusterAutoScaler)</td>
<td>Deactivated</td>
<td>You can deactivate Kubernetes Cluster Autoscaler. If you activate Kubernetes Cluster Autoscaler, a helm chart for this component is deployed.</td>
</tr>
</tbody>
</table>
### AWS Launch Wizard User Guide

#### Deploy to a new VPC

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFS storage class (EfsStorageClass)</td>
<td>Deactivated</td>
<td>You can activate deploying EFS storage to provide persistent storage that is redundant and untethered to individual Availability Zones.</td>
</tr>
<tr>
<td>Prometheus integration (PrometheusIntegration)</td>
<td>Deactivated</td>
<td>You can activate deploying Prometheus Helm charts into the Kubernetes cluster. For more information, see <a href="https://prometheus.io/">https://prometheus.io/</a>.</td>
</tr>
<tr>
<td>Grafana integration (GrafanaIntegration)</td>
<td>Deactivated</td>
<td>You can activate deploying Grafana Helm charts into the Kubernetes cluster. Grafana requires &quot;Prometheus integration&quot; to be enabled. For more information, see <a href="https://www.grafana.com/">https://www.grafana.com/</a>.</td>
</tr>
</tbody>
</table>

4. When you are satisfied with your infrastructure selections, choose **Next**. If you don't want to complete the configuration, choose **Cancel**. When you choose **Cancel**, all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, choose **Previous**.

5. After configuring your application, you are prompted to define the infrastructure requirements for the new deployment on the **Define infrastructure requirements** page. The following tabs provide information about the input fields.

**Compute**

- **Infrastructure requirements based on infrastructure.** You can choose to select your instances, or to use AWS recommended resources. If you choose to use AWS recommended resources, you have the option of defining your performance needs. If you don't select either option, default values are assigned.
- **Number of instance cores.** Choose the number of CPU cores for your infrastructure. The default value assigned is 4.
- **Network performance.** Choose your preferred network performance in Gbps.
- **Memory (GB).** Choose the amount of RAM that you want to attach to your EC2 instances. The default value assigned is 4 GB.
- **Recommended resources.** Launch Wizard displays the system-recommended resources based on your infrastructure selections. If you want to change the recommended resources, select different infrastructure requirements.
- **Infrastructure requirements based on instance type.** Choose to select your instance or to use AWS recommended resources. If you don't select either option, default values are assigned.
- **Instance type.** Select your preferred instance type from the dropdown list.

6. When you are satisfied with your infrastructure selections, select **Next**. If you don't want to complete the configuration, select **Cancel**. When you select **Cancel**, all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select **Previous**.

7. On the **Review and deploy** page, review your configuration details. If you want to make changes, select **Previous**. To stop, select **Cancel**. When you select **Cancel**, all of the selections on the specification page are lost and you are returned to the landing page. When you choose **Deploy**, you agree to the terms of the **Acknowledgment**. Launch Wizard validates the inputs and notifies you if you need to address any issues.
8. When validation is complete, Launch Wizard deploys your AWS resources and configures your Amazon EKS application. Launch Wizard provides you with status updates about the progress of the deployment on the Deployments page. From the Deployments page, you can view the list of current and previous deployments.

9. When your deployment is ready, a notification informs you that your Amazon EKS application is successfully deployed. If you have set up an Amazon SNS notification, you are also alerted through Amazon SNS. You can manage and access all of the resources related to your application by selecting the deployment, and then selecting Manage from the Actions dropdown list.

10. When the application is deployed, you can access your EC2 instances through the Amazon EC2 console.

**Deploy Amazon Elastic Kubernetes Service into an existing Amazon Virtual Private Cloud**

The following steps guide you through a Amazon EKS deployment with AWS Launch Wizard after you have launched it from the console.

1. When you select **Choose application** from the AWS Launch Wizard landing page, you are directed to the Choose application wizard where you are prompted to select the type of application that you want to deploy.

2. Select **Amazon EKS**, select **Deploy Amazon EKS into an existing VPC**, then select Create deployment.

3. You are prompted to enter the specifications for the new deployment. The following tabs provide information about the specification fields of the deployment model.

   **General**
   - **Deployment name**. Enter a unique application name for your deployment.
   - **Amazon Simple Notification Service (SNS) topic ARN — optional**. Specify an Amazon SNS topic where AWS Launch Wizard can send notifications and alerts. For more information, see the Amazon Simple Notification Service Developer Guide.
   - **Deactivate rollback on failed deployment**. By default, if a deployment fails, your provisioned resources will be deleted. You can enable this setting during deployment to prevent this behavior.
   - **Tags - optional**. Enter a key and value to assign metadata to your deployment. For help with tagging, see Tagging Your Amazon EC2 Resources.

   **Network configuration**
   - **Key pair name**. Select an existing key pair from the dropdown list or create a new one. If you select Create new key pair name, you are directed to the Amazon EC2 console. From there, under **Network and Security**, choose **Key Pairs**. Choose Create a new key pair, enter a name for the key pair, and then choose Download Key Pair.

   **Important**
   This is the only opportunity for you to save the private key file. Download it and save it in a safe place. You must provide the name of your key pair when you launch an instance and provide the corresponding private key each time that you connect to the instance. Return to the Launch Wizard console and choose the refresh button next to the Key Pairs dropdown list. The newly created key pair appears in the dropdown list. For more information about key pairs and Linux instances, see Amazon EC2 Key Pairs and Linux Instances. For more information about key pairs and Windows instances, see Amazon EC2 Key Pairs and Windows Instances.
• **Allowed external access CIDR**: Allowed CIDR block for external access to the deployed instances.

The following table shows all the input fields corresponding to the VPC, public subnets, and private subnets.

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPC ID (VPCID)</td>
<td>Requires input</td>
<td>ID of your existing VPC (for example, vpc-0343606e).</td>
</tr>
<tr>
<td>Private subnet 1 ID (PrivateSubnet1ID)</td>
<td>Requires input</td>
<td>ID of the private subnet in Availability Zone 1 of your existing VPC (for example, subnet-fe9a8b32).</td>
</tr>
<tr>
<td>Private subnet 2 ID (PrivateSubnet2ID)</td>
<td>Requires input</td>
<td>ID of the private subnet in Availability Zone 2 of your existing VPC (for example, subnet-be8b01ea).</td>
</tr>
<tr>
<td>Private subnet 3 ID (PrivateSubnet3ID)</td>
<td>Blank string</td>
<td>(Optional) ID of the private subnet in Availability Zone 3 of your existing VPC (for example, subnet-abd39039).</td>
</tr>
<tr>
<td>Public subnet 1 ID (PublicSubnet1ID)</td>
<td>Requires input</td>
<td>ID of the public subnet in Availability Zone 1 of your existing VPC (for example, subnet-a0246dcd).</td>
</tr>
<tr>
<td>Public subnet 2 ID (PublicSubnet2ID)</td>
<td>Requires input</td>
<td>ID of the public subnet in Availability Zone 2 of your existing VPC (for example, subnet-b1236eea).</td>
</tr>
<tr>
<td>Public subnet 3 ID (PublicSubnet3ID)</td>
<td>Blank string</td>
<td>(Optional) ID of the public subnet in Availability Zone 3 of your existing VPC (for example, subnet-c3456aba).</td>
</tr>
</tbody>
</table>

**VPC architecture requirements:**

- **VPC ID**: Amazon EC2 is hosted in multiple locations world-wide. These locations are composed of AWS Regions and Availability Zones. Amazon VPC enables you to launch AWS resources into a virtual network that you’ve defined. Choose the VPC that you want to use from the dropdown list. Your VPC must be associated at least two public subnets and two private subnets.

- **Availability Zone (AZ) configuration**: You must choose two or three Availability Zones in the Region. Each of the Availability Zones will have a private subnet and a public subnet in the selected VPC. A subnet is a range of IP addresses within a VPC that is allocated in an Availability Zone for the Region.

- **Public Subnets**: You must choose at least two public subnets for your VPC.
If a subnet’s traffic is routed to an internet gateway, it is a public subnet. If a subnet doesn’t have a route to the internet gateway, it is a private subnet. To use an existing VPC that does not have a public subnet, add a new public subnet using the following steps.

- Follow the steps in Creating a Subnet in the Amazon VPC User Guide using the existing VPC that you intend to use in AWS Launch Wizard.
- Add an internet gateway to your VPC, by following the steps in Attaching an Internet Gateway in the Amazon VPC User Guide.
- Configure your subnets to route internet traffic through the internet gateway, by following the steps in Creating a Custom Route Table in the Amazon VPC User Guide. Use IPv4 format (0.0.0.0/0) for the destination.
- Enable the required public subnet setting of auto-assign public IPv4 address. To enable this setting, follow the steps in Modifying the Public IPv4 Addressing Attribute for Your Subnet in the Amazon VPC User Guide.

**Important**
You must tag each public subnet being used with the key kubernetes.io/role/elb and the value true.

- **Private subnets:** You must choose at least two private subnets for your VPC.

If a subnet doesn’t have a route to an internet gateway, the subnet is known as a private subnet. To create a private subnet, you can use the following steps. We recommend that you enable the outbound connectivity for each of your selected private subnets using a NAT Gateway. To enable outbound connectivity from private subnets with public subnet, see the steps in Creating a NAT Gateway to create a NAT Gateway in your chosen public subnet. Then, follow the steps in Updating Your Route Table for each of your chosen private subnets.

- Follow the steps in Creating a Subnet in the Amazon VPC User Guide using the existing VPC you will use in AWS Launch Wizard.
- When you create a VPC, it includes a main route table by default. On the Route Tables page in the Amazon VPC console, you can view the main route table for a VPC by looking for Yes in the Main column. The main route table controls the routing for all subnets that are not explicitly associated with any other route table. If the main route table for your VPC has an outbound route to an internet gateway, then any subnet created using the previous step, by default, becomes a public subnet. To ensure that the subnets are private, you may need to create separate route tables for your private subnets. These route tables must not contain any routes to an internet gateway. Alternatively, you can create a custom route table for your public subnet and remove the internet gateway entry from the main route table.

**Important**
You must tag each private subnet being used with the key kubernetes.io/role/internal-elb and the value true.
## EKS configuration

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Config set name (ConfigSetName)</td>
<td>Blank string</td>
<td>(Optional) This parameter is used to map advanced parameters to an EKS cluster. You can keep it blank unless you are using an advanced configuration stack. If you launched an advanced configuration stack and want to apply its values to this cluster, this name must match the ConfigSetName parameter for the stack. If kept blank, a new config set is created using default values.</td>
</tr>
<tr>
<td>HTTP proxy (HttpProxy)</td>
<td>Blank string</td>
<td>(Optional) HTTP(S) proxy configuration. If provided, all worker nodes and pod egress traffic uses this proxy.</td>
</tr>
<tr>
<td>Per-account shared resources (PerAccountSharedResources)</td>
<td>AutoDetect</td>
<td>This EKS deployment creates several IAM roles and instance profiles that are intended to be deployed only once in an AWS account. If you already have an existing Launch Wizard EKS application deployed in this AWS account, in this AWS Region or another, you must choose No to skip creation of the per-account shared resources.</td>
</tr>
</tbody>
</table>
## Deploy to an existing VPC

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per-Region shared resources (PerRegionSharedResources)</td>
<td>AutoDetect</td>
<td>This EKS deployment sets up several resources such as helper Lambda functions, an Amazon S3 bucket for staging assets, and AWS CloudFormation macros that are intended to be deployed once for each AWS Region and shared in future deployments of Launch Wizard EKS in that Region. If you already have an existing Launch Wizard EKS application deployed in this account in this Region, you must choose <strong>No</strong> to skip creation of the per-Region shared resources.</td>
</tr>
<tr>
<td>Provision bastion host (ProvisionBastionHost)</td>
<td>Activated</td>
<td>Skip creating a bastion host by deactivating this option.</td>
</tr>
<tr>
<td>EKS cluster name (EKSClusterName)</td>
<td><strong>Blank string</strong></td>
<td>(Optional) Name for the EKS cluster. If kept blank, one is automatically generated. This must be unique within the Region.</td>
</tr>
<tr>
<td>EKS public access endpoint (EKSPublicAccessEndpoint)</td>
<td>Deactivated</td>
<td>Configure access to the Kubernetes API server endpoint from outside of your VPC.</td>
</tr>
<tr>
<td>Additional EKS admin ARN (IAM user) (AdditionalEKSAdminUserArn)</td>
<td><strong>Blank string</strong></td>
<td>(Optional) IAM user ARN to be granted administrative access to the EKS cluster.</td>
</tr>
<tr>
<td>Additional EKS admin ARN (IAM role) (AdditionalEKSAdminRoleArn)</td>
<td><strong>Blank string</strong></td>
<td>(Optional) IAM role ARN to be granted administrative access to the EKS cluster.</td>
</tr>
<tr>
<td>Fargate namespaces (FargateNamespaces)</td>
<td><strong>Blank string</strong></td>
<td>(Optional) Comma-separated list of namespaces for which Fargate should be enabled.</td>
</tr>
</tbody>
</table>

### EKS node group configuration

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes (NumberOfNodes)</td>
<td>3</td>
<td>Number of Amazon EKS node instances. The default is one for each of the three Availability Zones.</td>
</tr>
</tbody>
</table>
## Deploy to an existing VPC

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of nodes (MaxNumberOfNodes)</td>
<td>3</td>
<td>Maximum number of Amazon EKS node instances. The default is three.</td>
</tr>
<tr>
<td>Node group OS (NodeGroupOS)</td>
<td>Amazon Linux 2</td>
<td>Operating system to use for node instances. Choose Bottlerocket for the Amazon purpose-built container OS (unmanaged node groups only). Note that if you choose Windows, an additional Amazon Linux node group is created.</td>
</tr>
<tr>
<td>Node group type (NodeGroupType)</td>
<td>Managed</td>
<td>Choose Unmanaged to create an Auto Scaling group without using the EKS-managed node groups feature.</td>
</tr>
<tr>
<td>Node instance family (NodeInstanceFamily)</td>
<td>Standard</td>
<td>Choose the instance family to match the value of Node instance type.</td>
</tr>
</tbody>
</table>

### Kubernetes add-ins

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS load balancer controller (ALBIngressController)</td>
<td>Activated</td>
<td>You can deactivate deploying the AWS load balancer controller. If you activate deployment of the AWS load balancer controller, a Helm chart for this component is deployed.</td>
</tr>
<tr>
<td>Cluster autoscaler (ClusterAutoScaler)</td>
<td>Deactivated</td>
<td>You can deactivate Kubernetes Cluster Autoscaler. If you activate Kubernetes Cluster Autoscaler, a Helm chart for this component is deployed.</td>
</tr>
<tr>
<td>Amazon EFS storage class (EfsStorageClass)</td>
<td>Deactivated</td>
<td>You can activate deploying EFS storage to provide persistent storage that is redundant and untethered to individual Availability Zones.</td>
</tr>
<tr>
<td>Prometheus integration (PrometheusIntegration)</td>
<td>Deactivated</td>
<td>You can activate deploying Prometheus Helm charts into the Kubernetes cluster. For more information, see <a href="https://prometheus.io/">https://prometheus.io/</a>.</td>
</tr>
</tbody>
</table>
4. When you are satisfied with your infrastructure selections, select Next. If you don't want to complete the configuration, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select Previous.

5. After configuring your application, you are prompted to define the infrastructure requirements for the new deployment on the Define infrastructure requirements page. The following tabs provide information about the input fields.

**Compute**

- **Infrastructure requirements based on infrastructure.** You can choose to select your instances, or to use AWS recommended resources. If you choose to use AWS recommended resources, you have the option of defining your performance needs. If you don't select either option, default values are assigned.
- **Number of instance cores.** Choose the number of CPU cores for your infrastructure. The default value assigned is 4.
- **Network performance.** Choose your preferred network performance in Gbps.
- **Memory (GB).** Choose the amount of RAM that you want to attach to your EC2 instances. The default value assigned is 4 GB.
- **Recommended resources.** Launch Wizard displays the system-recommended resources based on your infrastructure selections. If you want to change the recommended resources, select different infrastructure requirements.
- **Infrastructure requirements based on instance type.** Choose to select your instance or to use AWS recommended resources. If you don't select either option, default values are assigned.
- **Instance type.** Select your preferred instance type from the dropdown list.

6. When you are satisfied with your infrastructure selections, select Next. If you don't want to complete the configuration, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select Previous.

7. On the Review and deploy page, review your configuration details. If you want to make changes, select Previous. To stop, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the landing page. When you choose Deploy, you agree to the terms of the Acknowledgment. Launch Wizard validates the inputs and notifies you if you need to address any issues.

8. When validation is complete, Launch Wizard deploys your AWS resources and configures your Amazon EKS application. Launch Wizard provides you with status updates about the progress of the deployment on the Deployments page. From the Deployments page, you can view the list of current and previous deployments.

9. When your deployment is ready, a notification informs you that your Amazon EKS application is successfully deployed. If you have set up an Amazon SNS notification, you are also alerted through Amazon SNS. You can manage and access all of the resources related to your application by selecting the deployment, and then selecting Manage from the Actions dropdown list.

10. When the application is deployed, you can access your Amazon EC2 instances through the Amazon EC2 Console.
Test the deployment

Note
You must run these steps from a network that has access to the Kubernetes API, as configured by the Amazon EKS public access endpoint and Kubernetes API public access CIDR parameters. For more information, see Installing kubectl in the Amazon EKS User Guide. If you enabled the optional bastion host, you can connect to it by using SSH. Use the key pair that you specified during deployment and the IP address from the Outputs tab of the AWS CloudFormation stack. The bastion host already has kubectl installed and configured so that it connects to the cluster. To test the CLI, connect to the cluster, and run the command, shown in step 1.

Follow this procedure to test the deployment:

1. Run the following command:

   ```
   # kubectl version
   ```

2. Confirm that the output includes the server version, which indicates a successful connection to the Kubernetes control plane.

   ```
   ```

3. Check for a successful connection between the nodes and cluster by running the `kubectl get nodes` command.

   ```
   # kubectl get nodes
   NAME STATUS ROLES AGE VERSION
   ip-10-0-25-239.us-west-2.compute.internal Ready <none> 10m <version number>
   ip-10-0-27-244.us-west-2.compute.internal Ready <none> 10m <version number>
   ip-10-0-35-29.us-west-2.compute.internal Ready <none> 10m <version number>
   ```

Best practices

Best practices for using Amazon EKS on AWS

Topics
- Amazon EKS application best practices (p. 40)
- Use AWS CloudFormation for ongoing management (p. 41)
- Monitor additional resource usage (p. 41)
- Security (p. 41)

Amazon EKS application best practices

For more information about best practices for your Amazon EKS application, see the EKS Best Practices Guides.
Use AWS CloudFormation for ongoing management

We recommend using CloudFormation for managing updates and resources that are created by this Launch Wizard deployment. Using the Amazon EC2 Console, AWS CLI, or API to change or delete resources can cause future CloudFormation operations on the stack to behave unexpectedly.

Monitor additional resource usage

This deployment enables users of the Amazon EKS cluster to use Elastic Load Balancing and Amazon EBS volumes as part of their Kubernetes applications. Because these carry additional costs, we recommend that you grant users of the Amazon EKS cluster the minimum permissions required according to Kubernetes Role Based Access Control (RBAC). We also recommend that you monitor resource usage by using the Kubernetes CLI or API to describe persistent volume claims (PVC) and Elastic Load Balancing resources across all namespaces. To disable this functionality, update the ControlPlaneRole IAM role in the child stack to restrict access to the Kubernetes control plane for specific AWS APIs, such as ec2:CreateVolume and elb:CreateLoadBalancer.

Security

Amazon EKS uses IAM to authenticate your Kubernetes cluster, but it still relies on native Kubernetes RBAC. This means that IAM is used only for valid entities. All permissions for interacting with your Amazon EKS cluster’s Kubernetes API are managed by the native Kubernetes RBAC system. We recommend that you grant least privilege access through Kubernetes RBAC.

Troubleshoot AWS Launch Wizard for Amazon EKS

Each application in your account in the same AWS Region can be uniquely identified by the application name specified at the time of a deployment. The application name can be used to view the details related to the application launch.

Contents

- Launch Wizard provisioning events (p. 41)
- AWS CloudFormation stack (p. 41)
- Application launch quotas (p. 42)
- Enable termination protection (p. 42)
- Errors (p. 43)

Launch Wizard provisioning events

Launch Wizard captures events from AWS CloudFormation to track the status of an ongoing application deployment. If an application deployment fails, you can access the AWS CloudFormation console to view the deployment events for this application by selecting Deployments from the navigation pane. A failed event shows a status of Failed along with a failure message.

AWS CloudFormation stack

Launch Wizard uses AWS CloudFormation to provision the infrastructure resources of an application. You can view the status of these AWS CloudFormation stacks, and if any of the stacks fail, you can view the cause of the failure. AWS CloudFormation stacks can be found in your account using the AWS CloudFormation describe-stacks API or by accessing the stack in the AWS CloudFormation console. The following can be used with the describe-stacks API for the --stack-name argument:
• Application resources

LaunchWizard-APPLICATION_NAME. This stack also has nested stacks for VPC, EKS control plane, node group, load balancer, and bastion hosts, among other components.

Application launch quotas

Launch Wizard allows three active applications with the status of in progress at one time. The combined maximum amount of in progress and completed active applications is 25 for any given application type. If you want to increase this limit, contact AWS Support.

Enable termination protection

If you encounter errors when you deploy Amazon EKS with Launch Wizard, and the log information provided by Launch Wizard or AWS CloudFormation is not sufficient to determine your issue, you must connect to the instance within the Amazon EC2 Auto Scaling group to determine the cause of the failure. When you connect to an instance to troubleshoot deployment failures, a common cause is the deployment scripts failing on the operating system. The following error messages in AWS CloudFormation can indicate the deployment scripts failed:

• Received 1 FAILURE signal(s) out of 1. Unable to satisfy 100% MinSuccessfulInstancesPercent requirement

• WaitCondition received failed message: ‘Error: Failed in function <script function name>. Return code 1 , warnings: <any warnings>’ for uniqueId: <Resource/wait condition name>

• <Resource name> timed out. Failed to receive 1 resource signal(s) within the specified duration

• Unparsable WaitCondition data

You can only connect to an EC2 instance if it is not terminated. Launch Wizard terminates instances on stack creation failure by default. You can enable the Deactivate rollback on failed deployment setting during deployment to prevent this behavior. If the setting was not enabled, you can still prevent the instance from getting terminated by updating the termination settings of that instance from the EC2 console before the AWS CloudFormation stack gets rolled back.

Note
When you enable Deactivate rollback on failed deployment, you continue to incur AWS charges for the stack. Ensure that you delete the stack when you finish troubleshooting.

To find the EC2 instances from the Launch Wizard deployment

2. Choose the AWS CloudFormation stack of the Launch Wizard deployment, and choose the Resources tab.
3. Choose the resource with type AWS::AutoScaling::AutoScalingGroup.
4. Select the instance management tab. This page will have a link to the EC2 console, which lists the instances in the Launch Wizard deployment.

You can update the termination settings to disable termination of the instances from the EC2 console. From the Instances page, select an instance and choose Action > Instance Settings > Change Termination Protection. Then choose Yes, Enable.
After you have determined the root cause, disable the termination protection before you delete the deployment in Launch Wizard.

Errors

Your requested instance type is not supported in your requested Availability Zone

- **Cause:** This failure might occur during the launch of either your EKS cluster instances or bastion hosts.
- **Solution:** You must choose a different Availability Zone and retry the deployment from the initial page of the Launch Wizard console.

EC2 instance stabilization error

- **Cause:** Failure can occur if an EC2 instance fails to stabilize. When this happens, the EC2 instance is unable to communicate to the AWS CloudFormation service to signal completions, resulting in WaitCondition errors.
- **Solution:** WaitCondition errors are often transient EC2 failures and retrying the deployment may succeed. For additional assistance, contact AWS Support.

Permission errors

- **Cause:** Insufficient IAM permissions could be the cause of various failures in the EKS deployment. Errors caused by insufficient permissions may occur within the EC2 instances as scripts are run during the application deployment. Other errors may return a verbose message indicating there are insufficient permissions similar to the following:

```plaintext
```

- **Solution:** Before deploying the Launch Wizard application, you must sign in to the AWS Management Console with IAM permissions for the resources that Launch Wizard will deploy. The AdministratorAccess managed policy within IAM provides sufficient permissions, although your organization may choose to use a custom policy with more restrictions.
What is AWS Launch Wizard for Remote Desktop Gateway?

AWS Launch Wizard for Remote Desktop Gateway (RD Gateway) guides you through the sizing, configuration, and deployment of RD Gateway on the AWS Cloud. RD Gateway uses the Remote Desktop Protocol (RDP) over HTTPS to establish a secure, encrypted connection between remote users and Amazon Elastic Compute Cloud instances running Windows, without needing to configure a virtual private network (VPN). This helps reduce the attack surface on your Windows instances while providing a remote administration solution for administrators.

Contents

- Deployment options (p. 44)
- Features (p. 45)
- Components (p. 46)

Deployment options

This Launch Wizard application provides two deployment options:

- **Deploy RD Gateway into a new VPC (end-to-end deployment).** Builds a new AWS environment consisting of a VPC, subnets, NAT gateways, security groups, and other infrastructure components, and then deploys RD Gateway into this new VPC.

- **Deploy RD Gateway into an existing VPC.** Provisions standalone RD Gateway instances in your existing AWS infrastructure.

AWS Launch Wizard provides separate templates for these two deployment types. You can also configure CIDR blocks, instance types, and RD Gateway settings.
Features

AWS Launch Wizard provides the following features:

- Simple application deployment (p. 45)
- Application Resource Groups for discoverability (p. 45)
- AWS resource selection (p. 45)
- Cost estimation (p. 45)
- SNS notification (p. 45)
- Early input validation (p. 45)

Simple application deployment

AWS Launch Wizard makes it more efficient for you to deploy third-party applications on AWS, such as Remote Desktop Gateway. When you input the application requirements, AWS Launch Wizard deploys the necessary AWS resources for a production-ready application. This means that you do not have to manage separate infrastructure pieces or spend as much time provisioning and configuring your Remote Desktop Gateway application.

Application Resource Groups for discoverability

Launch Wizard creates a Resource Group for all of the AWS resources created for your Remote Desktop Gateway application. You can manage the resources through the Amazon EC2 Console or with AWS Systems Manager. When you access Systems Manager through Launch Wizard, the resources are automatically filtered for you based on your Resource Group. You can manage, patch, and maintain your Remote Desktop Gateway applications in Systems Manager.

AWS resource selection

Launch Wizard considers performance, memory, bandwidth, and other application features to determine the most appropriate instance type for your Remote Desktop Gateway application. You can modify the recommended defaults.

Cost estimation

Launch Wizard provides a cost estimate for a complete deployment. The cost estimate is itemized for each individual resource to deploy. The estimated cost automatically updates each time you change a resource type configuration in the wizard. The provided estimates are for general comparisons only. The estimates are based on On-Demand costs and actual costs may be lower.

SNS notification

You can provide an Amazon SNS topic so that Launch Wizard will send you notifications and alerts about the status of a deployment.

Early input validation

Launch Wizard performs the following resource limit validations at the AWS account level:

- VPC
- Internet gateway
Components

An RD Gateway application deployed with Launch Wizard includes the following components:

- A highly available architecture that spans two Availability Zones.
- In each public subnet, up to four RD Gateway instances in an Auto Scaling group to provide secure remote access to instances in the private subnets. Each instance is assigned an Elastic IP address so it's reachable directly from the internet.
- A Network Load Balancer to provide RDP access to the RD Gateway instances.
- A security group for Windows instances that will host the RD Gateway role, with an ingress rule permitting TCP port 3389 from your administrator IP address. After deployment, you'll modify the security group ingress rules to configure administrative access through TCP port 443 instead.
- An empty application tier for instances in private subnets. If more tiers are required, you can create additional private subnets with unique CIDR ranges.
- AWS Systems Manager Parameter Store to securely store credentials used for accessing the RD Gateway instances.
- AWS Systems Manager to automate the deployment of the RD Gateway Auto Scaling group.
- Self-signed SSL certificate and configuration of Remote Desktop Connection Authorization Policies (RD CAPs) and RD Gateway.
- Resource Groups that contain all the resources created with Launch Wizard.

Additionally, a new VPC deployment includes the following components:

- A VPC configured with public and private subnets according to AWS best practices, to provide you with your own virtual network on AWS.
- An internet gateway to allow access to the internet. This gateway is used by the RD Gateway instances to send and receive traffic.
- Managed network address translation (NAT) gateways to allow outbound internet access for resources in the private subnets.
Get Started with AWS Launch Wizard for Remote Desktop Gateway

This section contains information to help you set up your environment to deploy RD Gateway with Launch Wizard. When your environment is set up, you can deploy RD Gateway application with Launch Wizard by following the steps and parameter specification details provided in this section.

Topics to help you get started:
- Access AWS Launch Wizard (p. 48)
• Set up (p. 48)
• Deploy standalone Remote Desktop Gateway into a new Amazon Virtual Private Cloud (p. 49)
• Deploy standalone Remote Desktop Gateway into an existing Amazon Virtual Private Cloud (p. 52)

Access AWS Launch Wizard

You can launch AWS Launch Wizard from the AWS Launch Wizard console located at https://console.aws.amazon.com/launchwizard.

Set up

The following prerequisites must be verified to deploy a Remote Desktop Gateway application with AWS Launch Wizard.
• Specialized knowledge (p. 48)
• Amazon Web Services account (p. 48)
• Service Quotas (p. 48)
• Amazon Elastic Compute Cloud key pairs (p. 49)
• AWS Identity and Access Management permissions (p. 49)

Specialized knowledge

This deployment requires a moderate level of familiarity with AWS services. If you’re new to AWS, see Getting Started Resource Center and AWS Training and Certification. These sites provide materials for learning how to design, deploy, and operate your infrastructure and applications on the AWS Cloud.

This Launch Wizard assumes familiarity with Remote Desktop Gateway.

Amazon Web Services account

If you don’t already have an AWS account, create one at http://aws.amazon.com/ by following the on-screen instructions. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.

Your AWS account is automatically signed up for all AWS services. You are charged only for the services you use.

Service Quotas

If necessary, request service quota increases for the following resources. You might need to request increases if your existing deployment currently uses these resources and if this Launch Wizard deployment could result in exceeding the default quotas. The Service Quotas console displays your usage and quotas for some aspects of some services. For more information, see What is Service Quotas? and AWS service quotas.

Existing VPC Service Quotas:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default quota</th>
<th>This deployment uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic IP Addresses</td>
<td>5 per Region</td>
<td>2</td>
</tr>
<tr>
<td>AWS Identity and Access Management (IAM) security groups</td>
<td>300 per account</td>
<td>1</td>
</tr>
</tbody>
</table>
Deploy to a new VPC

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default quota</th>
<th>This deployment uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAM roles</td>
<td>1,000 per account</td>
<td>1</td>
</tr>
<tr>
<td>Auto Scaling groups</td>
<td>200 per Region</td>
<td>1</td>
</tr>
<tr>
<td>Amazon EC2 On-Demand Instances (Standard)</td>
<td>5 per Region</td>
<td>1-4</td>
</tr>
</tbody>
</table>

New VPC Service Quotas:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default quota</th>
<th>This deployment uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPCs</td>
<td>5 per Region</td>
<td>1</td>
</tr>
<tr>
<td>Elastic IP Addresses</td>
<td>5 per Region</td>
<td>2</td>
</tr>
<tr>
<td>Internet Gateway</td>
<td>5 per Region</td>
<td>1</td>
</tr>
<tr>
<td>AWS Identity and Access Management (IAM) security groups</td>
<td>300 per account</td>
<td>1</td>
</tr>
<tr>
<td>IAM roles</td>
<td>1,000 per account</td>
<td>1</td>
</tr>
<tr>
<td>Auto Scaling groups</td>
<td>200 per Region</td>
<td>1</td>
</tr>
<tr>
<td>Amazon EC2 On-Demand Instances (Standard)</td>
<td>5 per Region</td>
<td>1-4</td>
</tr>
</tbody>
</table>

Amazon Elastic Compute Cloud key pairs

Ensure that at least one Amazon EC2 key pair exists in your AWS account in the Region where you plan to deploy the Launch Wizard application. Note the key pair name because you will use it during deployment. To create a key pair, see Amazon EC2 key pairs and Windows instances.

For testing or proof-of-concept purposes, we recommend creating a new key pair instead of using one that’s already being used by a production instance.

AWS Identity and Access Management permissions

Before deploying the Launch Wizard application, you must sign in to the AWS Management Console with IAM permissions for the resources that the templates deploy. The AdministratorAccess managed policy within IAM provides sufficient permissions, although your organization may choose to use a custom policy with more restrictions. For more information, see AWS managed policies for job functions.

Deploy standalone Remote Desktop Gateway into a new Amazon Virtual Private Cloud

The following steps guide you through a Remote Desktop Gateway deployment with AWS Launch Wizard after you have launched it from the console.

1. When you select **Choose application** from the AWS Launch Wizard landing page, you are directed to the Choose application wizard where you are prompted to select the type of application that you want to deploy.
2. Select Microsoft Remote Desktop Gateway, select Deploy into a new VPC, then select Create deployment.

3. You are prompted to enter the specifications for the new deployment. The following tabs provide information about the specification fields of the deployment model.

General

- **Deployment name**. Enter a unique application name for your deployment.
- **Amazon Simple Notification Service (SNS) topic ARN — optional**. Specify an Amazon SNS topic where AWS Launch Wizard can send notifications and alerts. For more information, see the Amazon Simple Notification Service Developer Guide.
- **Deactivate rollback on failed deployment**. By default, if a deployment fails, your provisioned resources will be deleted. You can enable this setting during deployment to prevent this behavior.
- **Tags - optional**. Enter a key and value to assign metadata to your deployment. For help with tagging, see Tagging Your EC2 Resources.

Network configuration

- **Key pair name**. Select an existing key pair from the dropdown list or create a new one. If you select Create new key pair name, you are directed to the Amazon EC2 console. From there, under Network and Security, choose Key Pairs. Choose Create a new key pair, enter a name for the key pair, and then choose Download Key Pair.

  **Important**
  This is the only opportunity for you to save the private key file. Download it and save it in a safe place. You must provide the name of your key pair when you launch an instance and provide the corresponding private key each time that you connect to the instance. Return to the Launch Wizard console and choose the refresh button next to the Key Pairs dropdown list. The newly created key pair appears in the dropdown list. For more information about key pairs, see Amazon EC2 Key Pairs and Windows Instances.

- **Availability Zone (AZ) configuration**: You must choose at least two Availability Zones. Deployment will create a highly available architecture that spans these Availability Zones.
- **VPC Settings**: Launch Wizard creates your VPC in this case. The following shows input fields that define VPC configuration.

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPC tenancy</td>
<td>default</td>
<td>The allowed tenancy of instances launched into the VPC.</td>
</tr>
<tr>
<td>VPC CIDR</td>
<td>10.0.0.0/16</td>
<td>CIDR block for the VPC.</td>
</tr>
<tr>
<td>Private subnet 1 CIDR</td>
<td>10.0.0.0/19</td>
<td>CIDR block for private subnet 1 located in Availability Zone 1.</td>
</tr>
<tr>
<td>Private subnet 2 CIDR</td>
<td>10.0.32.0/19</td>
<td>CIDR block for private subnet 2 located in Availability Zone 2.</td>
</tr>
<tr>
<td>Public subnet 1 CIDR</td>
<td>10.0.128.0/20</td>
<td>CIDR Block for the public DMZ subnet 1 located in Availability Zone 1.</td>
</tr>
</tbody>
</table>
Deploy to a new VPC

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public subnet 2 CIDR</td>
<td>10.0.144.0/20</td>
<td>CIDR Block for the public DMZ subnet 2 located in Availability Zone 2.</td>
</tr>
<tr>
<td>Allowed Remote Desktop Gateway external access CIDR</td>
<td>Requires input</td>
<td>Allowed CIDR block for external access to the Remote Desktop Gateways.</td>
</tr>
</tbody>
</table>

Microsoft Remote Desktop Gateway configuration

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of RDGW hosts</td>
<td>1</td>
<td>Enter the number of Remote Desktop Gateway hosts to create.</td>
</tr>
<tr>
<td>Admin user name</td>
<td>StackAdmin</td>
<td>User name for the new local administrator account.</td>
</tr>
<tr>
<td>Admin password</td>
<td>Requires input</td>
<td>Password for the administrative account. Must be at least 8 characters containing letters, numbers, and symbols.</td>
</tr>
</tbody>
</table>

4. When you are satisfied with your infrastructure selections, select **Next**. If you don't want to complete the configuration, select **Cancel**. When you select **Cancel**, all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select **Previous**.

5. After configuring your application, you are prompted to define the infrastructure requirements for the new deployment on the **Define infrastructure requirements** page. The following tabs provide information about the input fields.

**Compute**

- **Infrastructure requirements based on instance type.** You can choose to select your instances or use AWS recommended resources. If you choose to use AWS recommended resources, you have the option of defining your performance needs. If no selections are made, default values are assigned.
- **Number of instance cores.** Choose the number of CPU cores for your infrastructure. The default value assigned is 4.
- **Network performance.** Choose your preferred network performance in Gbps.
- **Memory (GB).** Choose the amount of RAM that you want to attach to your EC2 instances. The default value assigned is 4 GB.
- **Recommended resources.** Launch Wizard displays the system-recommended resources based on your infrastructure selections. If you want to change the recommended resources, select different infrastructure requirements.
- **Infrastructure requirements based on instance type.** You can choose to select your instance or use AWS recommended resources. If no selections are made, default values are assigned.
- **Instance type.** Select your preferred instance type from the dropdown list.

6. When you are satisfied with your infrastructure selections, select **Next**. If you don't want to complete the configuration, select **Cancel**. When you select **Cancel**, all of the selections on the specification...
page are lost and you are returned to the landing page. To go to the previous screen, select Previous.

7. On the Review and deploy page, review your configuration details. If you want to make changes, select Previous. To stop, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the landing page. When you choose Deploy, you agree to the terms of the Acknowledgment. Launch Wizard validates the inputs and notifies you of any issues you must address.

8. When validation is complete, Launch Wizard deploys your AWS resources and configures your Microsoft Remote Desktop Gateway application. Launch Wizard provides you with status updates about the progress of the deployment on the Deployments page. From the Deployments page, you can view the list of current and previous deployments.

9. When your deployment is ready, a notification informs you that your Remote Desktop Gateway application is successfully deployed. If you have set up an Amazon SNS notification, you are also alerted through Amazon SNS. To manage and access all of the resources related to your application, select the deployment, and from the Actions dropdown list, select Manage.

10. When the application is deployed, you can access your EC2 instances through the Amazon EC2 console.

Deploy standalone Remote Desktop Gateway into an existing Amazon Virtual Private Cloud

The following steps guide you through a Remote Desktop Gateway deployment with AWS Launch Wizard after you have launched it from the console.

1. When you select Choose application from the AWS Launch Wizard landing page, you are directed to the Choose application wizard where you are prompted to select the type of application that you want to deploy.

2. Select Microsoft Remote Desktop Gateway, select Deploy into an existing VPC, then select Create deployment.

3. You are prompted to enter the specifications for the new deployment. The following tabs provide information about the specification fields of the deployment model.

   General

   - Deployment name. Enter a unique application name for your deployment.
   - Amazon Simple Notification Service (SNS) topic ARN — optional. Specify an Amazon SNS topic where AWS Launch Wizard can send notifications and alerts. For more information, see the Amazon Simple Notification Service Developer Guide.
   - Deactivate rollback on failed deployment. By default, if a deployment fails, your provisioned resources will be deleted. You can enable this setting during deployment to prevent this behavior.
   - Tags - optional. Enter a key and value to assign metadata to your deployment. For help with tagging, see Tagging Your Amazon EC2 Resources.

   Network configuration

   Key pair name. Select an existing key pair from the dropdown list or create a new one. If you select Create new key pair name, you are directed to the Amazon EC2 console. From there, under Network and Security, choose Key Pairs. Choose Create a new key pair, enter a name for the key pair, and then choose Download Key Pair.
Important
This is the only opportunity for you to save the private key file. Download it and save it in a safe place. You must provide the name of your key pair when you launch an instance and provide the corresponding private key each time that you connect to the instance. Return to the Launch Wizard console and choose the refresh button next to the Key Pairs dropdown list. The newly created key pair appears in the dropdown list. For more information about key pairs, see Amazon EC2 Key Pairs and Windows instances.

VPC Settings:

- Select Virtual Private Cloud (VPC) option. Choose the VPC that you want to use from the dropdown list. Your VPC must be associated at least two public subnets for HA deployments.

To add a new public subnet

If a subnet's traffic is routed to an internet gateway, the subnet is known as a public subnet. If, however, a subnet doesn't have a route to the internet gateway, the subnet is known as a private subnet. To use an existing VPC that does not have a public subnet, you can add a new public subnet using the following steps.

- Follow the steps in Creating a Subnet in the Amazon VPC User Guide using the existing VPC you intend to use AWS Launch Wizard.
- To add an internet gateway to your VPC, follow the steps in Attaching an Internet Gateway in the Amazon VPC User Guide.
- To configure your subnets to route internet traffic through the internet gateway, follow the steps in Creating a Custom Route Table in the Amazon VPC User Guide. Use IPv4 format (0.0.0.0/0) for Destination.
- The public subnet should have the “auto-assign public IPv4 address” setting enabled. To enable this setting, follow the steps in Modifying the Public IPv4 Addressing Attribute for Your Subnet in the Amazon VPC User Guide.

- Availability Zone (AZ) configuration: You must choose at least two Availability Zones. The deployment will create a highly available architecture that spans these Availability Zones.

- Allowed Remote Desktop Gateway external access CIDR: You must specify a CIDR block for allowing external RDP access to the Remote Desktop Gateways on TCP port 3389.

Microsoft Remote Desktop Gateway configuration

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of RDGW hosts</td>
<td>1</td>
<td>Enter the number of Remote Desktop Gateway hosts to create.</td>
</tr>
<tr>
<td>Admin user name</td>
<td>StackAdmin</td>
<td>User name for the new local administrator account.</td>
</tr>
<tr>
<td>Admin password</td>
<td>Requires input</td>
<td>Password for the administrative account. Must be at least 8 characters containing letters, numbers, and symbols.</td>
</tr>
</tbody>
</table>

4. When you are satisfied with your infrastructure selections, select Next. If you don’t want to complete the configuration, select Cancel. When you select Cancel, all of the selections on the specification
After configuring your application, you are prompted to define the infrastructure requirements for the new deployment on the **Define infrastructure requirements** page. The following tabs provide information about the input fields.

**Compute**

- **Infrastructure requirements based on instance type.** You can choose to select your instances, or to use AWS recommended resources. If you choose to use AWS recommended resources, you have the option of defining your performance needs. If no selections are made, default values are assigned.
- **Number of instance cores.** Choose the number of CPU cores for your infrastructure. The default value assigned is 4.
- **Network performance.** Choose your preferred network performance in Gbps.
- **Memory (GB).** Choose the amount of RAM that you want to attach to your EC2 instances. The default value assigned is 4 GB.
- **Recommended resources.** Launch Wizard displays the system-recommended resources based on your infrastructure selections. If you want to change the recommended resources, select different infrastructure requirements.
- **Infrastructure requirements based on instance type.** You can choose to select your instance or use AWS recommended resources. If no selections are made, default values are assigned.
- **Instance type.** Select your preferred instance type from the dropdown list.

6. When you are satisfied with your infrastructure selections, select **Next.** If you don't want to complete the configuration, select **Cancel.** When you select **Cancel,** all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select **Previous.**

7. On the **Review and deploy** page, review your configuration details. If you want to make changes, select **Previous.** To stop, select **Cancel.** When you select **Cancel,** all of the selections on the specification page are lost and you are returned to the landing page. When you choose **Deploy,** you agree to the terms of the **Acknowledgment.** Launch Wizard validates the inputs and notifies you of any issues you must address.

8. When validation is complete, Launch Wizard deploys your AWS resources and configures your **Microsoft Remote Desktop Gateway** application. Launch Wizard provides you with status updates about the progress of the deployment on the **Deployments** page. From the **Deployments** page, you can view the list of current and previous deployments.

9. When your deployment is ready, a notification informs you that your **Remote Desktop Gateway** application is successfully deployed. If you have set up an Amazon SNS notification, you are also alerted through Amazon SNS. You can manage and access all of the resources related to your application by selecting the deployment, and then selecting **Manage** from the **Actions** dropdown list.

10. When the application is deployed, you can access your EC2 instances through the Amazon EC2 console.

---

**Post-deployment steps**

Post-deployment steps for Remote Desktop Gateway on AWS

**Topics**

- Complete the configuration of your AWS environment (p. 55)
- Install the root certificate (p. 55)
Complete the configuration of your AWS environment

After AWS Launch Wizard finishes the application deployment, follow these steps:

1. Create security groups for your Windows instances that will be located in private VPC subnets. Create an ingress rule permitting TCP port 3389 from the RD Gateway security group, CIDR range, or IP address. Associate these groups with instances as they are launched into the private subnets.

2. Make sure that your administrative clients can resolve the name for the RD Gateway endpoint (for example, win-1a2b3c4d5e6.example.com). You can create an A (Host) record in DNS that maps the FQDN to the RD gateway's Elastic IP or public IP address. For testing purposes, you can configure this mapping in the local host's file on the machine.

3. Configure administrative clients with the proper configuration settings. This includes installing the root certificate from each RD Gateway server on the client machines (see the next section for instructions). When you use the CloudFormation templates, the default location for the root certificate will be `c:\servername.cer` on each RD Gateway server.

4. Modify the RD Gateway security group. Remove the ingress rule permitting TCP port 3389. Create a new ingress rule permitting TCP port 443 from your administrator's IP address.

5. Make sure that instances in private subnets are associated with a security group containing ingress rules permitting the RD Gateway server IP address to connect through TCP port 3389.

6. Configure the Remote Desktop connection for administrative clients, as described later in this section.

Install the root certificate

This Launch Wizard deployment implements a self-signed certificate on the RD gateway instances. After deployment, you must install the root certificate on your administrative clients before you configure the RDP client to connect to your RD gateway instances. The root certificate will automatically be stored as `c:\servername.cer`.

To distribute this file to administrator workstations and install it, follow these steps:

1. Open a Command Prompt window using administrative credentials.
2. Type `mmc` and press Enter.
3. In the Console Root window, on the File menu, choose Add/Remove Snap In.
4. In the Add Standalone Snap-in dialog box, choose Certificates, and then choose Add.
5. In the Certificates snap-in dialog box, choose Computer account, and then choose Next.
6. In the Select Computer dialog box, choose Finish.
7. In the Add Standalone Snap-in dialog box, choose Close.
8. On the Add/Remove Snap-in dialog box, choose OK.
9. In the Console Root window, expand Certificates (Local Computer).
10. Under Certificates (Local Computer), expand Trusted Root Certification Authorities.
11. Open the context (right-click) window for Certificates, and choose All Tasks > Import.
12. Navigate to the root certificate (e.g., RDGW1.cer) to complete the installation.
Configure the Remote Desktop Connection Client

1. Start the Remote Desktop Connection client.
2. In the computer name field, type the name or IP address of the Windows instance you want to connect to. Keep in mind that this instance needs to be reachable only from the RD gateway, not from the client machine.


4. Choose Use these RD Gateway server settings. For server name, specify the FQDN of the RD gateway. If the RD gateway and the server you want to connect to are in the same domain, choose Use my RD Gateway credentials for the remote computer, and then choose OK.

Note
The root certificate will be stored as c:\servername.cer on each RD gateway when deploying servers using the CloudFormation templates.
Configure the Remote Desktop Connection Client

Note
The FQDN server name of the RD Gateway host must match the certificate and the DNS record (or local HOSTS file entry). Otherwise, the secure connection will generate warnings and might fail.

5. Enter your credentials, and then choose OK to connect to the server. You can supply the same set of credentials for the RD gateway and the destination server, as shown. If your servers are not joined to the domain, you will need to authenticate twice: once for the RD gateway and once for the destination server. If your servers aren't joined to the domain, when prompted for the RD Gateway server credentials, provide the Admin User Name and Admin Password credentials you set in when you deployed with Launch Wizard. Check the Remember my credentials box. (Otherwise, if you’re connecting from a Windows computer, you'll get prompted for your credentials repeatedly, and will be blocked from entering your remote computer credentials.)
Run Windows Updates

In order to ensure the deployed servers’ operating systems and installed applications have the latest Microsoft updates, run Windows Update on each server.

1. Create an RDP session to the Remote Desktop Gateway server(s).
2. Open the Settings application.
4. Click Check for updates.
5. Install any updates and reboot if necessary.

Best practices

Best practices for using Remote Desktop Gateway on AWS

Topics

- The Principle of Least Privilege (p. 59)
- VPC Configuration (p. 59)
- Network Access Control Lists (p. 59)
- Security groups (p. 60)
- Initial Remote Administration Architecture (p. 61)
- SSL Certificates (p. 62)
- Connection and Resource Authorization Policies (p. 63)
The Principle of Least Privilege

When considering remote administrative access to your environment, it is important to follow the principle of least privilege. This principle refers to users having the fewest possible permissions necessary to perform their job functions. This helps reduce the attack surface of your environment, making it much harder for an adversary to exploit. An attack surface can be defined as the set of exploitable vulnerabilities in your environment, including the network, software, and users who are involved in the ongoing operation of the system.

Following the principle of least privilege, we recommend reducing the attack surface of your environment by exposing the absolute minimal set of ports to the network while also restricting the source network or IP address that will have access to your Amazon Elastic Compute Cloud instances.

In addition to the functionality that exists in the Windows platform, there are several AWS capabilities to help you implement the principle of least privilege, such as subnets, security groups, and trusted ingress CIDR blocks.

VPC Configuration

Amazon Virtual Private Cloud lets you provision a private, isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. With Amazon VPC, you can define a virtual network topology closely resembling a traditional network that you might operate on your own premises. You control your virtual networking environment. This includes the selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

When deploying Windows architecture on the AWS Cloud, we recommend a VPC configuration that supports the following requirements:

- Place critical workloads in a minimum of two Availability Zones to provide high availability.
- Place instances into individual tiers. For example, in a Microsoft SharePoint deployment, you should have separate tiers for web servers, application servers, database servers, and domain controllers. Traffic between these groups can be controlled to adhere to the principle of least privilege.
- Deploy RD Gateways into public subnets in each Availability Zone for remote administration. Other components, such as reverse proxy servers, can also be placed into these public subnets if needed.

Network Access Control Lists

A network access control list (ACL) is a set of permissions that you can attach to any network subnet in a VPC to provide stateless filtering of traffic. You can use network ACLs for inbound or outbound traffic, as they provide an effective way to place a CIDR block or individual IP addresses on a deny list. These ACLs can contain ordered rules to allow or deny traffic based on IP protocol, service port, or source or destination IP address. The following image shows the default ACL configuration for a VPC subnet, which is also used by this Launch Wizard deployment:
You can keep the default network ACL configuration, or you can configure more specific rules to restrict traffic between subnets at the network level. For example, you could set a rule that would allow inbound administrative traffic on TCP port 3389 from a specific set of IP addresses. In either case, you must implement security group rules to permit access from users connecting to RD Gateways and between tiered groups of Amazon EC2 instances.

### Security groups

All instances are required to belong to one or more security groups. Security groups allow you to set policies to control open ports and provide isolation between application tiers. In a VPC, every instance runs behind a stateful firewall with all ports closed by default. The security group contains rules responsible for opening inbound and outbound ports on that firewall. While security groups act as an instance-level firewall, they can also be associated with multiple instances, providing isolation between application tiers in your environment. For example, you can create a security group for all your web servers that will allow traffic on TCP port 3389, but only from members of the security group containing your RD Gateway servers. The following diagram illustrates this configuration:

Notice that inbound connections from the internet are only permitted over TCP port 443 to the RD Gateways. The RD Gateways have an Elastic IP address assigned and have direct access to the internet. The remaining Windows instances are deployed into private subnets and are assigned private IP addresses only. Security group rules allow only the RD Gateways to initiate inbound connections for remote administration to TCP port 3389 for instances in the private subnets.
In this architecture, RDP connections are established over HTTPS to the RD Gateway and proxied to backend instances on the standard RDP TCP port 3389. This configuration helps you reduce the attack surface on your Windows instances while allowing administrators to establish connections to all your instances through a single gateway.

It’s possible to provide remote administrative access to all your Windows instances through one RD Gateway, but we recommend placing gateways in each Availability Zone for redundancy. This Launch Wizard deployment implements this best practice for you.

Initial Remote Administration Architecture

In an initial RD Gateway configuration, the servers in the public subnet will need an inbound security group rule permitting TCP port 3389 from the administrator's source IP address or subnet. Windows instances sitting behind the RD Gateway in a private subnet will be in their own isolated tier. For example, a group of web server instances in a private subnet may be associated with their own web tier security group. This security group will need an inbound rule allowing connections from the RD Gateway on TCP port 3389.

Using this architecture, an administrator can use a traditional RDP connection to an RD Gateway to configure the local server. The RD Gateway can also be used as a bastion host (jump box). This means that when an RDP connection is established to the desktop of the RD Gateway, an administrator can start a new RDP client session to initiate a connection to an instance in a private subnet, as illustrated in the following diagram:

Although this architecture works well for initial administration, it is not recommended for the long term. To further secure connections and reduce the number of RDP sessions required to administer the servers in the private subnets, the inbound rule should be changed to permit TCP port 443. The RD Gateway service should be installed and configured with an SSL certificate and Remote Desktop Connection Authorization Policies (RD CAP).

This Launch Wizard deployment sets up a standard TCP port 3389 connection from the administrator’s IP address. You must follow the post-deployment steps to modify the security group for RD Gateway to use a single inbound rule permitting TCP port 443. This modification will allow a Transport Layer Security (TLS) encrypted RDP connection to be proxied through the gateway over TCP port 443 directly to one or more Windows instances in private subnets on TCP port 3389. This configuration increases the security of the connection and also prevents the need to initiate an RDP session to the desktop of the RD Gateway. The following diagram illustrates this configuration:
SSL Certificates

The RD Gateway role uses Transport Layer Security (TLS) to encrypt communications over the internet between administrators and gateway servers. To support TLS, a valid X.509 SSL certificate must be installed on each RD Gateway. Certificates can be acquired in a number of ways, including:

- Your own PKI infrastructure, such as a Microsoft Enterprise Certificate Authority (CA)
- Certificates issued by a public CA, such as Verisign or Digicert
- Self-signed certificates

For smaller test environments, implementing a self-signed certificate is a straightforward process that helps you get up and running quickly. This Launch Wizard deployment automatically generates a self-signed certificate for RD Gateway.

However, if you have a large number of varying administrative devices that need to establish a connection to your gateways, we recommend using a public certificate.

For an RDP client to establish a secure connection with an RD Gateway, the following certificate and DNS requirements must be met:

- The issuing CA of the certificate installed on the gateway must be trusted by the RDP client. For example, the root CA certificate must be installed in the client machine’s Trusted Root Certification Authorities store.
- The subject name used on the certificate installed on the gateway must match the DNS name used by the client to connect to the server; for example, rdgw1.example.com.
- The client must be able to resolve the hostname (for example, rdgw1.example.com) to the Elastic IP address of the RD Gateway. This will require a Host (A) record in DNS.

There are various considerations when choosing the right CA to obtain an SSL certificate. For example, a public certificate may be ideal, because the issuing CA will be widely trusted by the majority of client devices that need to connect to your gateways. However, you may want to use your own PKI infrastructure to ensure that only the machines that are part of your organization will trust the issuing CA.
Connection and Resource Authorization Policies

Users must meet specific requirements to connect to RD Gateway instances:

- **Connection Authorization Policies** – Remote Desktop Connection Authorization Policies (RD CAPs) allow you to specify who can connect to an RD Gateway instance. For example, you can select a group of users from your domain, such as Domain Admins.

- **Resource Authorization Policies** – Remote Desktop Resource Authorization Policies (RD RAPs) allow you to specify the internal Windows instances that remote users can connect to through an RD Gateway instance. For example, you can choose specific computers joined to a domain, which administrators can connect to through the RD Gateway.

This Launch Wizard deployment automatically sets up Connection and Resource Authorization Policies.

Troubleshoot AWS Launch Wizard for Remote Desktop Gateway

Each application in your account in the same AWS Region can be uniquely identified by the application name specified at the time of a deployment. The application name can be used to view the details related to the application launch.

Contents

- Launch Wizard provisioning events (p. 63)
- AWS CloudFormation stack (p. 63)
- Application launch quotas (p. 64)
- Enable termination protection (p. 64)
- Errors (p. 65)

Launch Wizard provisioning events

Launch Wizard captures events from AWS CloudFormation to track the status of an ongoing application deployment. If an application deployment fails, you can access the AWS CloudFormation console to view the deployment events for this application by selecting **Deployments** from the navigation pane. A failed event shows a status of **Failed** along with a failure message.

AWS CloudFormation stack

Launch Wizard uses AWS CloudFormation to provision the infrastructure resources of an application. You can view the status of these AWS CloudFormation stacks, and if any of the stacks fail, you can view the cause of the failure. AWS CloudFormation stacks can be found in your account using the AWS CloudFormation **describe-stacks** API or by accessing the stack in the AWS CloudFormation console. The following can be used with the **describe-stacks** API for the **--stack-name** argument:

- **Application resources**

  *LaunchWizard--APPLICATION_NAME*. This stack also has nested stacks for VPC and the RDGW node.
Application launch quotas

Launch Wizard allows three active applications with the status of in progress at one time. The combined maximum amount of in progress and completed active applications is 25 for any given application type. If you want to increase this limit, contact AWS Support.

Enable termination protection

If you encounter errors when you deploy Remote Desktop Gateway with Launch Wizard, and the log information provided by Launch Wizard or AWS CloudFormation is not sufficient to determine your issue, you must connect to the instance within the Amazon EC2 Auto Scaling group to determine the cause of the failure. When you connect to an instance to troubleshoot deployment failures, a common cause is the deployment scripts failing on the operating system. The following error messages in AWS CloudFormation can indicate that the deployment scripts failed:

- Received 1 FAILURE signal(s) out of 1. Unable to satisfy 100% MinSuccessfulInstancesPercent requirement
- WaitCondition received failed message: ‘Error: Failed in function <script function name>. Return code 1 , warnings: <any warnings>’ for uniqueId: <Resource/wait condition name>
- <Resource name> timed out. Failed to receive 1 resource signal(s) within the specified duration
- Unparsable WaitCondition data

You can only connect to an EC2 instance if it is not terminated. Launch Wizard terminates instances on stack creation failure by default. You can enable the Deactivate rollback on failed deployment setting during deployment to prevent this behavior. If the setting was not enabled, you can still prevent the instance from getting terminated by updating the termination settings of that instance from the EC2 console before the AWS CloudFormation stack gets rolled back.

**Note**

When you enable Deactivate rollback on failed deployment, you continue to incur AWS charges for the stack. Ensure that you delete the stack when you finish troubleshooting.

To find the EC2 instances from the Launch Wizard deployment

2. Choose the AWS CloudFormation stack of the Launch Wizard deployment, and choose the Resources tab.
3. Choose the resource with type AWS::AutoScaling::AutoScalingGroup.
4. Select the instance management tab. This page will have a link to the EC2 console, which lists the instances in the Launch Wizard deployment.

You can update the termination settings to disable termination of the instances from the EC2 console. From the Instances page, select an instance and choose Action > Instance Settings > Change Termination Protection. Then choose Yes, Enable.

After you have determined the root cause, disable the termination protection before you delete the deployment in Launch Wizard.
Errors

Your requested instance type is not supported in your requested Availability Zone

- **Cause:** This failure might occur during the launch of your RD Gateway instances.
- **Solution:** You must choose a different Availability Zone and retry the deployment from the initial page of the Launch Wizard console.

EC2 instance stabilization error

- **Cause:** Failure can occur if an EC2 instance fails to stabilize. When this happens, the EC2 instance is unable to communicate to the AWS CloudFormation service to signal completions, resulting in WaitCondition errors.
- **Solution:** WaitCondition errors are often transient EC2 failures and retrying the deployment may succeed. For additional assistance, contact AWS Support.

Permission errors

- **Cause:** Insufficient IAM permissions could be the cause of various failures in the RD Gateway deployment. Errors caused by insufficient permissions may occur within the EC2 instances as scripts are run during the application deployment. Other errors may return a verbose message indicating there are insufficient permissions similar to the following:

  
  ```
  User: arn:aws:iam::123456789098:user/test-user is not authorized to perform:
  elasticloadbalancing:CreateTargetGroup on resource: arn:aws:elasticloadbalancing:us-east-1:123456789098:targetgroup/myTargetGroup/*
  ```

- **Solution:** Before deploying the Launch Wizard application, you must sign in to the AWS Management Console with IAM permissions for the resources that Launch Wizard will deploy. The AdministratorAccess managed policy within IAM provides sufficient permissions, although your organization may choose to use a custom policy with more restrictions.
AWS Launch Wizard for SAP

This section of the AWS Launch Wizard documentation contains guidance specific to the deployment of SAP applications on AWS using the Launch Wizard service.

Topics

- What is AWS Launch Wizard for SAP? (p. 66)
- How AWS Launch Wizard for SAP works (p. 77)
- Get started with AWS Launch Wizard for SAP (p. 82)
- Manage application resources with AWS Launch Wizard for SAP (p. 108)
- Make SAP HANA software available for AWS Launch Wizard to deploy a HANA database (p. 109)
- Make SAP application software available for AWS Launch Wizard to deploy SAP (p. 110)
- Repeat SAP application deployments using deployment artifacts created with AWS Launch Wizard (p. 118)
- Scale SAP applications with AWS Launch Wizard for SAP after initial deployment (p. 141)
- Security groups in AWS Launch Wizard for SAP (p. 148)
- Troubleshoot AWS Launch Wizard for SAP (p. 150)

What is AWS Launch Wizard for SAP?

AWS Launch Wizard for SAP is a service that guides you through the sizing, configuration, and deployment of SAP applications on AWS, and follows AWS cloud application best practices.

AWS Launch Wizard reduces the time it takes to deploy SAP applications on AWS. You input your application requirements, including SAP HANA settings, SAP landscape settings, and deployment details on the service console, and Launch Wizard identifies the appropriate AWS resources to deploy and run your SAP application. Launch Wizard provides an estimated cost of deployment, which allows you to modify your resources and instantly view the updated cost. When you finalize your settings, Launch Wizard provisions and configures the selected resources. It then optionally installs an SAP HANA database and supported SAP applications using customer-provided software.

After you deploy an SAP application, you can access it from the Amazon EC2 console. You can manage your SAP applications with AWS SSM.

Contents

- Supported deployments and features of AWS Launch Wizard (p. 66)
- Supported versions for SAP deployments (p. 69)
- Components (p. 75)
- Related services (p. 76)

Supported deployments and features of AWS Launch Wizard

Supported deployments

AWS Launch Wizard currently supports the deployment of AWS resources for the following SAP systems and patterns. SAP HANA database software and supported SAP application software are optionally installed and provided by the customer.
**AWS Launch Wizard User Guide**

**Supported deployments and features**

- **SAP HANA database on a single Amazon EC2 instance.** Deploy SAP HANA in a single-node, scale-up architecture, with up to 24TB of memory.
- **SAP NetWeaver on SAP HANA system on a single Amazon EC2 instance.** Deploy an SAP application on the same Amazon EC2 instance as your SAP HANA Database.
- **SAP HANA database on multiple EC2 instances.** Deploy SAP HANA in a multi-node, scale-out architecture.
- **SAP NetWeaver system on multiple EC2 instances.** Deploy an SAP NetWeaver system using a distributed deployment model, which includes an ASCS/PAS server, single/multiple SAP HANA servers running SAP HANA databases, and multiple application servers.
- **Cross-AZ SAP HANA database high availability setup.** Deploy SAP HANA with high availability configured across two Availability Zones.
- **Cross-AZ SAP NetWeaver system setup.** Deploy Amazon EC2 instances for ASCS/ERS and SAP HANA databases across two Availability Zones, and spread the deployment of application servers across them.
- **SUSE/RHEL cluster setup** For SAP HANA and NetWeaver on HANA high availability deployments, Launch Wizard for SAP configures SUSE/RHEL clustering when you provide SAP software and specify the deployment of SAP database or application software. For SAP HANA databases, clustering is enabled between the ASCS and ERS nodes.

**AWS Launch Wizard provides the following features:**

- Instance selection and configuration (p. 67)
- AWS resource selection (p. 67)
- Cost estimation (p. 67)
- Reusable infrastructure settings (p. 68)
- SNS notification (p. 68)
- Application resource groups (p. 68)
- AWS Data Provider for SAP (p. 68)
- AWS Backint Agent for SAP HANA (p. 68)
- Custom deployment configuration scripts (p. 68)
- Application software installation (p. 69)
- Creation of AWS Service Catalog products (p. 69)

**Instance selection and configuration**

When you input the application requirements, Launch Wizard deploys the necessary AWS resources for a production-ready application. This means that you do not have to figure out how to select the right instances and configure them to run supported SAP applications.

**AWS resource selection**

Launch Wizard considers CPU/Memory or SAPS requirements that you provide to determine the most appropriate instance types and other resources for your SAP application. You can modify the recommended defaults.

**Cost estimation**

Launch Wizard provides a cost estimate for the complete deployment that is itemized for each individual resource being deployed. The estimated cost automatically updates each time you change a resource type configuration in the wizard. The provided estimates are only for general comparisons. They are based on On-Demand Instance costs. Actual costs may be lower.
Reusable infrastructure settings

You can save the settings for your AWS infrastructure for the SAP landscape to reuse when you want to deploy SAP systems that function similarly within a landscape. For example, a development configuration can be created for the first development instance, which can later be reused to deploy other development systems.

Some example scenarios for which DevOps and SAP architecture teams can create templates include:

- Organize the SAP systems within a landscape.
- Save infrastructure settings, including VPC, subnets, key pairs, and security groups to ensure that systems that must be deployed with the same settings are correctly deployed.
- Set up connectivity between the systems using the same configuration template so they can communicate with each other when security groups are created with Launch Wizard.
- Use the same GID for SAPSYS group across different configuration templates to ensure that SAP transport files systems are mounted properly.

SNS notification

You can provide an SNS topic so that Launch Wizard will send you notifications and alerts about the status of a deployment.

Application resource groups

Launch Wizard creates a resource group for all of the AWS resources created for your SAP system. You can manage the resources through the Amazon EC2 console or by using Systems Manager.

AWS Data Provider for SAP

Deploying and running the Amazon Web Services (AWS) Data Provider for SAP is a prerequisite for running SAP systems on AWS. Launch Wizard automatically deploys AWS Data Provider for SAP on every Amazon EC2 instance that it launches. AWS Data Provider for SAP is a tool that collects performance-related data from AWS services. It makes this data available to SAP applications to help monitor and improve the performance of business transactions. AWS Data Provider for SAP uses operating system, network, and storage data that is most relevant to the operation of the SAP infrastructure. Its data sources include Amazon EC2 and Amazon CloudWatch.

**Note**

AWS Data Provider is installed using a downloadable installer, and is not automatically updated to the latest version. You must manually check for and deploy updates, or manually set up an SSM distributor to install and update this service as described in Installing DataProvider 4.1.

AWS Backint Agent for SAP HANA

Launch Wizard deploys and configures AWS Backint Agent for SAP HANA, an SAP-certified backup and restore application for SAP HANA workloads running on Amazon EC2 instances in the cloud. Launch Wizard supports the deployment and configuration of Backint Agent for single-node, multi-node, and high availability deployments for supported HANA and NetWeaver on HANA applications.

Custom deployment configuration scripts

You can provide custom pre-deployment and post-deployment configuration scripts that can run on various instance tiers, such as SAP HANA Database, Primary Application Server, and Enqueue Replication Server during the pre-deployment and post-deployment configuration phases. Launch Wizard uses a
standalone component manager application (AWSTOE) to run the scripts. For more information, see Custom deployment configuration scripts (p. 81).

**Application software installation**

Launch Wizard can install SAP application software that you have made available on Amazon S3, including SAP NetWeaver on HANA, SAP S/4HANA, and SAP BW/4HANA. For more details about which operating systems and database versions are supported for each deployment pattern, see Supported application software installation versions and deployment patterns (p. 71). For supported software versions and installation details, see Make SAP application software available for AWS Launch Wizard to deploy SAP (p. 110).

**Creation of AWS Service Catalog products**

AWS Launch Wizard can create AWS Service Catalog products from successful deployments. The AWS Service Catalog products contain AWS CloudFormation templates and associated application configuration scripts, which are stored in Amazon S3. You can use the AWS Service Catalog products, along with integrations offered by AWS Service Catalog, with third-party products, such as ServiceNow, Jira, or Terraform. Or, you can use the AWS CloudFormation templates and application configuration scripts saved in Amazon S3 to deploy SAP applications that meet the requirements of organizational deployment and governance policies.

In addition to supporting deployments using AWS CloudFormation templates, AWS Service Catalog, and multiple deployment tools supported by AWS Service Catalog, AWS Launch Wizard creates a point-in-time snapshot of the code used to deploy and configure SAP applications at the time of the deployment. You can use the code in its current form for consistent repeated deployments, or you can use the code as a baseline and update it to meet specific application requirements.

**Supported versions for SAP deployments**

**Supported versions**

- Supported operating system versions for SAP deployments (p. 69)
- Supported application versions for SAP deployments (p. 71)
- Supported application software installation versions and deployment patterns (p. 71)

**Supported operating system versions for SAP deployments**

The following table provides details for the operating systems supported by Launch Wizard for SAP deployments.

<table>
<thead>
<tr>
<th>Operating system version</th>
<th>Single-node deployment</th>
<th>ASCS</th>
<th>ERS</th>
<th>PAS</th>
<th>SAP HANA database</th>
<th>SAP HANA database in HA cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red-Hat-Enterprise-Linux-8.2-For-SAP-HA-US-HVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

69
### Version support for SAP deployments

<table>
<thead>
<tr>
<th>Operating system version</th>
<th>Single-node deployment</th>
<th>ASCS</th>
<th>ERS</th>
<th>PAS</th>
<th>SAP HANA database</th>
<th>SAP HANA database in HA cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red-Hat-Enterprise-Linux-7.6-For-SAP-HA-US-HVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SuSE-Linux-15-SP2-HVM</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SuSE-Linux-15-SP2-For-SAP-HVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SuSE-Linux-15-SP2-For-SAP-BYOS-HVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SuSE-Linux-15-SP1-HVM</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SuSE-Linux-15-SP1-For-SAP-HVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SuSE-Linux-15-SP1-For-SAP-BYOS-HVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SuSE-Linux-15-HVM</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SuSE-Linux-15-For-SAP-HVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SuSE-Linux-15-For-SAP-BYOS-HVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SuSE-Linux-12-SP5-HVM</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Supported application versions for SAP deployments

The following table provides details for the application versions supported by Launch Wizard for SAP deployments.

<table>
<thead>
<tr>
<th>Application name</th>
<th>Supported versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP NetWeaver on HANA</td>
<td>7.5 SPS02 and 7.5 SPS00</td>
</tr>
<tr>
<td>SAP BW/4HANA</td>
<td>2.0 and 2021</td>
</tr>
<tr>
<td>SAP S/4HANA</td>
<td>1909, 2020, and 2021</td>
</tr>
</tbody>
</table>

Supported application software installation versions and deployment patterns

The following table provides details about the supported application software installation versions for the deployment patterns supported by Launch Wizard for SAP.

<table>
<thead>
<tr>
<th>Operating system version</th>
<th>Single-node deployment</th>
<th>ASCS</th>
<th>ERS</th>
<th>PAS</th>
<th>SAP HANA database</th>
<th>SAP HANA database in HA cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>SuSE-Linux-12-SP5-For-SAP-HVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SuSE-Linux-12-SP5-For-SAP-BYOS-HVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SuSE-Linux-12-SP4-HVM</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SuSE-Linux-12-SP4-For-SAP-HVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SuSE-Linux-12-SP4-For-SAP-BYOS-HVM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

For Additional Application Server (AAS), the operating system is inherited from the operating system selected for the PAS server.
<table>
<thead>
<tr>
<th>Application and software version</th>
<th>OS</th>
<th>HANA DB versions</th>
<th>Single instance deployment</th>
<th>Distributed instance deployment with HANA scale-up</th>
<th>Distributed instance deployment with HANA scale-out</th>
<th>High availability deployment*</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetWeaver 7.5 SP502</td>
<td>SLES 15 SP2</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP502</td>
<td>SLES 15 for SAP SP2</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP502</td>
<td>SLES 15 SP1</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP502</td>
<td>SLES 15 for SAP SP1</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP502</td>
<td>SLES 15</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP502</td>
<td>SLES 15 for SAP</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP502</td>
<td>SLES 12 SP5</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP502</td>
<td>SLES 12 for SAP SP5</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP502</td>
<td>RHEL 8.2 for SAP</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP502</td>
<td>RHEL 8.1 for SAP</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP502</td>
<td>RHEL 7.6 for SAP</td>
<td>HANA 2.0 SPS 05 &amp;</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Application and software version

<table>
<thead>
<tr>
<th>Application and software version</th>
<th>OS</th>
<th>HANA DB versions</th>
<th>Single instance deployment</th>
<th>Distributed instance deployment with HANA scale-up</th>
<th>Distributed instance deployment with HANA scale-out</th>
<th>High availability deployment*</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetWeaver 7.5 SP5</td>
<td>SLES 15 SP2</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP5</td>
<td>SLES 15 for SAP SP2</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP5</td>
<td>SLES 15 SP1</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP5</td>
<td>SLES 15 for SAP SP1</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP5</td>
<td>SLES 15</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP5</td>
<td>SLES 15 for SAP</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP5</td>
<td>SLES 12 SP5</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP5</td>
<td>SLES 12 for SAP SP5</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP5</td>
<td>RHEL 8.2 for SAP</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>NetWeaver 7.5 SP5</td>
<td>RHEL 8.1 for SAP</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Application and software version</td>
<td>OS</td>
<td>HANA DB versions</td>
<td>Single instance deployment</td>
<td>Distributed instance deployment with HANA scale-up</td>
<td>Distributed instance deployment with HANA scale-out</td>
<td>High availability deployment*</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----</td>
<td>------------------</td>
<td>---------------------------</td>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>NetWeaver 7.5 SPS0</td>
<td>RHEL 7.6 for SAP</td>
<td>HANA 2.0 SPS 05 &amp; HANA 2.0 SPS 04</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S4HANA 2020 &amp; 1909</td>
<td>SLES 15 SP2</td>
<td>HANA 2.0 SPS 05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>S4HANA 2020 &amp; 1909</td>
<td>SLES 15 for SAP SP2</td>
<td>HANA 2.0 SPS 05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S4HANA 2020 &amp; 1909</td>
<td>SLES 15 SP1</td>
<td>HANA 2.0 SPS 05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>S4HANA 2020 &amp; 1909</td>
<td>SLES 15 for SAP SP1</td>
<td>HANA 2.0 SPS 05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S4HANA 2020 &amp; 1909</td>
<td>SLES 15 for SAP</td>
<td>HANA 2.0 SPS 05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>S4HANA 2020 &amp; 1909</td>
<td>SLES 15 for SAP</td>
<td>HANA 2.0 SPS 05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S4HANA 2020 &amp; 1909</td>
<td>SLES 12 SP5</td>
<td>HANA 2.0 SPS 05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>S4HANA 2020 &amp; 1909</td>
<td>SLES 12 for SAP SP5</td>
<td>HANA 2.0 SPS 05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S4HANA 2020 &amp; 1909</td>
<td>RHEL 8.2 for SAP</td>
<td>HANA 2.0 SPS 05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S4HANA 2020 &amp; 1909</td>
<td>RHEL 8.1 for SAP</td>
<td>HANA 2.0 SPS 05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>S4HANA 2020 &amp; 1909</td>
<td>RHEL 7.6 for SAP</td>
<td>HANA 2.0 SPS 05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>BW4HANA 2.0</td>
<td>SLES 15 SP2</td>
<td>HANA 2.0 SPS 05</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
### Components

An SAP application deployed with Launch Wizard includes the following components.

**SAP applications:**

- **SAP HANA Database** supports the following:
  - single instance deployment
  - distributed instance deployment in a single Availability Zone
  - cross-Availability Zone, high-availability deployment

- **SAP applications based on NetWeaver on SAP HANA database** supports the following:
  - single instance deployment
  - distributed instance deployment
  - cross-Availability Zone, high-availability deployment

**Security groups**

Launch Wizard creates optional security groups to ensure that all of the systems sharing the same configuration template can communicate with each other and with systems and end users who access...
the SAP systems from an IP CIDR range, an external IP address, or security groups. For more information about how Launch Wizard creates security groups and how they are configured, see Security groups in AWS Launch Wizard for SAP (p. 148).

**SAP transport group configuration**

You can create an SAP transport file system, or attach an existing transport file system that was created as part of a previous deployment with AWS Launch Wizard. Transport file systems are created with Amazon Elastic File System. For more information, see Amazon Elastic File System setup for transport directory (p. 79).

**Related services**

The following AWS services are used when you deploy an SAP application with AWS Launch Wizard.

**Services**

- AWS CloudFormation (p. 76)
- Amazon Virtual Private Cloud security groups (p. 76)
- Amazon Elastic File System (p. 76)
- Amazon EC2 Systems Manager (p. 76)
- Amazon Simple Notification Service (SNS) (p. 77)
- Amazon Route 53 (p. 77)
- AWS Backint Agent for SAP HANA (p. 77)
- AWS Task Orchestrator and Executor (p. 77)

**AWS CloudFormation**

AWS CloudFormation is a service that helps you model and set up your AWS resources, and lets you spend more time focusing on your applications that run in AWS. You create a template that describes all of the AWS resources that you want (for example, Amazon EC2 instances or Amazon RDS DB instances), and AWS CloudFormation takes care of provisioning and configuring those resources for you. With AWS Launch Wizard for SAP, you don't need to build AWS CloudFormation templates to deploy your application. Instead, AWS Launch Wizard combines infrastructure provisioning and application configuration (code that runs on EC2 instances to configure the application) into a unified AWS CloudFormation template. The AWS CloudFormation template is then invoked by AWS Launch Wizard's backend service to provision an application in your account.

**Amazon Virtual Private Cloud security groups**

Amazon Virtual Private Cloud security groups act as a virtual firewall for your instance to control inbound and outbound traffic. When you launch an instance in a VPC, you can assign up to five security groups to the instances. AWS Launch Wizard displays the security groups that will be assigned to the EC2 instances that run the SAP applications. This allows the components to communicate.

**Amazon Elastic File System**

Amazon EFS provides file storage in the AWS Cloud. With Amazon EFS, you can create a file system, mount the file system on an Amazon EC2 instance, and then read and write data to and from your file system. For more information, see Amazon Elastic File System setup for transport directory (p. 79).

**Amazon EC2 Systems Manager**

Amazon EC2 Systems Manager is an AWS service that you can use to view and control your infrastructure on AWS. Using the Amazon EC2 Systems Manager console, you can view operational data from multiple AWS services and automate operational tasks across your AWS resources. SSM helps you maintain
security and compliance by scanning your managed instances and reporting on, or taking corrective action on, any policy violations that it detects.

**Amazon Simple Notification Service (SNS)**

Amazon Simple Notification Service (SNS) is a highly available, durable, secure, fully managed pub/sub messaging service that provides topics for high-throughput, push-based, many-to-many messaging. Using Amazon SNS topics, your publisher systems can fan out messages to a large number of subscriber endpoints and send notifications to end users using mobile push, SMS, and email. You can use SNS topics for your Launch Wizard deployments to stay up-to-date on deployment progress. For more information, see the Amazon Simple Notification Service Developer Guide.

**Amazon Route 53**

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking. Launch Wizard integrates with Route 53 hosted zones, which are containers for records. The records contain information about how you want to route traffic for a specific domain, such as example.com, and its subdomains (acme.example.com, zenith.example.com). There are two types of hosted zones: public and private hosted zones. We recommend that you use private hosted zones for SAP applications unless an application must be directly accessible from the internet.

**AWS Backint Agent for SAP HANA**

AWS Backint Agent for SAP HANA is an SAP-certified backup and restore application for SAP HANA workloads running on Amazon EC2 instances in the cloud. AWS Backint Agent runs as a standalone application that integrates with your existing workflows to back up your SAP HANA database to Amazon S3 and to restore it using SAP HANA Cockpit, SAP HANA Studio, and SQL commands. AWS Backint Agent supports full, incremental, and differential backup of SAP HANA databases. Additionally, you can back up log files and catalogs to Amazon S3. AWS Backint Agent runs on an SAP HANA database server, where backups and catalogs are transferred from the SAP HANA database to the AWS Backint Agent. The AWS Backint Agent stores your files in the S3 bucket that is specified in the agent configuration file. To restore your SAP HANA database server, SAP HANA reads the catalog files stored in your S3 bucket using the AWS Backint Agent. It then initiates a request to restore the required files from S3.

**AWS Task Orchestrator and Executor**

AWS Task Orchestrator and Executor is component management application used to orchestrate complex workflows, modify system configurations, and test your systems without writing code. This application uses a declarative document schema. As a standalone application it does not require additional server setup. It can run on any cloud infrastructure and on premises. AWS Launch Wizard uses this application to orchestrate the download of the pre- and post-configuration scripts, and to run them.

**How AWS Launch Wizard for SAP works**

AWS Launch Wizard provisions and configures the infrastructure required to run SAP HANA database- and NetWeaver-based SAP applications on AWS. You select the SAP deployment pattern and provide the specifications, such as operating system, instance size, and vCPU/memory. Or, Launch Wizard can make these selections for you according to SAP Standard Application Benchmarks (SAPS). You have the option to manually choose the instance. Based on your selections, Launch Wizard automatically provisions the necessary AWS resources in the cloud.

Launch Wizard recommends Amazon EC2 instances by evaluating the SAPS or vCPU/memory requirements against the performance of Amazon EC2 instances supported by AWS. When new EC2 instances are released and certified for SAP, the sizing feature of Launch Wizard will take them into consideration when proposing recommendations.
Launch Wizard maintains a mapping rule engine built on the list of certified EC2 instances that are supported by SAP. When you enter your vCPU/memory or SAPS requirements, Launch Wizard recommends an Amazon EC2 instance that is certified for SAP workloads and offers performance that is no less than your input requirements. For certain workloads, such as SAP HANA in a production environment, Launch Wizard recommends instances based on the official SAP recommendations for SAP HANA database workloads. For workloads in a non-production environment, Launch Wizard recommends Amazon EC2 instances that meet SAP recommended requirements; however, the recommended instances are not enforced. You can change the instance types after deployment, or you can override the recommendation by making manual selections.

In addition to launching instances based on the SAP system information that you provide, such as SAP System Number and SAP System Identifier (SAP SID), Launch Wizard performs the following operations:

- Configures the operating system
- Configures hostname
- Attaches security groups so that the systems in the cluster that use the same configuration template, and also external systems, can communicate with the SAP systems that will be deployed on these instances.

Launch Wizard provides an estimated cost of deployment. You can modify your resources and instantly view an updated cost assessment. After you approve the deployment, Launch Wizard validates the inputs and flags inconsistencies. After you resolve the inconsistencies, Launch Wizard provisions and configures the resources. The result is a ready-to-use SAP application.

Launch Wizard creates a CloudFormation stack according to your infrastructure needs. For more information, see Working With Stacks in the AWS CloudFormation User Guide.

**How Launch Wizard for SAP works topics:**

- Implementation Details (p. 78)

## Implementation Details

AWS Launch Wizard implements SAP deployments as follows.

**Deployment aspects**

- Storage for SAP Systems (p. 78)
- Amazon Elastic File System setup for transport directory (p. 79)
- Amazon Elastic File System setup for SAP Central Services instances configured for high availability (p. 79)
- Bring your own image (BYOI) (p. 80)
- Specify private IP address (p. 80)
- Configuration settings (p. 81)
- Custom deployment configuration scripts (p. 81)
- Manual cleanup activities (p. 82)
- Default Quotas (p. 82)
- AWS Regions and Endpoints (p. 82)

## Storage for SAP Systems

Storage capacity and performance are key aspects of any SAP system installation. Launch Wizard provides storage type options for the SAP NetWeaver Application tier and the SAP HANA database tiers.
Amazon Elastic Block Store (Amazon EBS) volumes are included in the architecture to provide durable, high-performance storage. Amazon EBS volumes are network-attached disk storage, which you can create and attach to EC2 instances. When attached, you can create a file system on top of these volumes, run a database, or use them in any way that you would use a block device. Amazon EBS volumes are placed in a specific Availability Zone, where they are automatically replicated to protect you from the failure of a single component.

General Purpose EBS Volumes offer storage for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 16,000 IOPS (at 5,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size.

Provisioned IOPS Amazon EBS volumes offer storage with consistent and low-latency performance. They are backed by solid state drives (SSDs) and designed for applications with I/O intensive workloads, such as databases. Amazon EBS-optimized instances, such as the R4 instance type, deliver dedicated throughput between Amazon EC2 and Amazon EBS.

By default, Launch Wizard deploys Amazon EBS volumes for the SAP HANA database that meet the storage KPIs for SAP as listed in Storage Configuration for SAP HANA.

For NetWeaver database stacks, you can choose between a gp2, io1, or io2 volume for the /usr/sap/SAPSID and /sapmnt (for non-HA deployment architectures) file systems, whereas other configurations are deployed with gp2 volumes.

In an SAP landscape, development occurs in the development system and is then imported into the QA and follow-on systems. For this import to occur successfully, a shared file system is required for SAP systems in the landscape. Amazon EFS is used to create the SAP Transport file system that is shared between multiple SAP systems in the landscape.

Amazon Elastic File System setup for transport directory

The SAP transport directory is a shared file system between SAP systems (for example, Development, Quality, and Production) that are part of the same SAP Transport Domain for releasing and importing SAP transports. To avoid a single point of failure, Launch Wizard creates a file system with Amazon Elastic File System or reuses existing file systems. It mounts the file systems on the SAP systems that you select based on the role of the system. The transport file system is mounted on all of the applications servers included in the deployment.

When systems within the same SAP Transport Domain are created in one VPC and need to be attached to SAP systems in other VPCs (for example, if Development and Quality are deployed in a VPC tagged as Non_Prod, and Production is deployed in a VPC tagged as Prod), a prerequisite for using VPC Peering/Transit Gateway is that you must enable the VPCs to be able to communicate. This allows Launch Wizard to attach the transport directory created in one VPC to instance(s) in other VPCs using a mount target in the same Availability Zone or other Availability Zones, as applicable. If the VPCs are not permitted to communicate, then the deployment will fail when it attempts to mount the transport file system created in one VPC to systems in another VPC.

Note
When a transport files system is created with Amazon Elastic File System, Launch Wizard considers it a shared resource and will not delete it when you delete the deployment or if the deployment is rolled back.

Amazon Elastic File System setup for SAP Central Services instances configured for high availability

The SAP Central Services instances that make up a NetWeaver high availability deployment, ABAP Central Server (ASCS) and Enqueue Replication Server (ERS) instances, must contain the following file
systems to be highly available: `/sapmnt`, `/usr/sap/<SAPSID>/ASCS<XX>`, and `/usr/sap/<SAPSID>/ERS<XX>`. These file systems are built with Amazon EFS to avoid a single point of failure for the SAP system. Launch Wizard creates these file systems for the NetWeaver high availability pattern using a single Amazon Elastic File System.

The following table contains information about how a single Amazon EFS is configured and mounted on an ASCS, ERS, Primary Application Server (PAS), and Additional Application Server (AAS).

<table>
<thead>
<tr>
<th>EFS ID</th>
<th>EFS DNS name</th>
<th>Instance mounted on</th>
<th>File System name</th>
<th>Server mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>fs-123A456B</td>
<td>fs-123A456B.efs.&lt;AWS Region&gt;.amazonaws.com:/&gt;SAPMNT-&lt;SAPSID&gt;</td>
<td>/sapmnt</td>
<td>/sapmnt</td>
<td>SAP ASCS, ERS, Primary Application and Additional Application servers</td>
</tr>
<tr>
<td>fs-123A456B</td>
<td>fs-123A456B.efs.&lt;AWS Region&gt;.amazonaws.com:/ASCS-&lt;SAPSID&gt;</td>
<td>/usr/sap/&lt;SAPSID&gt;/ASCS&lt;XX&gt;</td>
<td>SAP ASCS Server</td>
<td></td>
</tr>
<tr>
<td>fs-123A456B</td>
<td>fs-123A456B.efs.&lt;AWS Region&gt;.amazonaws.com:/ERS-&lt;SAPSID&gt;</td>
<td>/usr/sap/&lt;SAPSID&gt;/ERS&lt;XX&gt;</td>
<td>SAP ERS Server</td>
<td></td>
</tr>
</tbody>
</table>

**Bring your own image (BYOI)**

You can bring your own images to deploy and configure EC2 instances for SAP with AWS Launch Wizard. During launch, in order to continue with a deployment, Launch Wizard verifies whether the operating system version selected on the front end matches the operating system version of the instance. If the versions do not match, the deployment fails with an error.

When building your own image, consider the following:

- Launch Wizard configures the operating systems with OS-level parameters and utilities required by SAP
- Refer to SAP installation documents to ensure that operating system prerequisites are in place so that Launch Wizard deployments do not fail.
- Launch Wizard accesses standard repositories provided by OS vendors. Do not block access to them.
- Deployments by Launch Wizard use OS utilities and programs, such as zipper, yum, grep, printf, awk, sed, autosfs, python, saptune, and tuned-profiles in the deployment script to configure SAP application and database servers. We recommend that you do not delete standard utilities.

**Specify private IP address**

You can specify available IP addresses that are already approved by your internal security and governance for each Amazon EC2 instance in your SAP deployment. The SAP environment is accessible as soon as the deployment is successful.

Launch Wizard, by default, auto-selects available IP addresses when a custom IP address is not provided.

When specifying a custom IP address, verify that it is within the range of the subnet of the instance that you are deploying.
Configuration settings

The following configuration settings are applied when deploying an SAP application with Launch Wizard.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Applies to</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSM Agent</td>
<td>All SAP systems and patterns</td>
</tr>
<tr>
<td>EBS volumes for SAP application tier</td>
<td>All SAP systems and patterns</td>
</tr>
<tr>
<td>EBS volumes for SAP HANA database, log and backup file systems</td>
<td>All SAP systems and patterns</td>
</tr>
<tr>
<td>EFS volumes for /hana/shared and /backup</td>
<td>All SAP systems and patterns</td>
</tr>
<tr>
<td>EFS volumes for SAP transport file systems</td>
<td>All SAP systems and patterns</td>
</tr>
<tr>
<td>EFS volumes for SAP central services: sapmnt, /usr/sap/&lt;SID&gt;/ASCS&lt;XX&gt;, and /usr/sap/&lt;SID&gt;/ERS&lt;XX</td>
<td>ASCS and ERS systems</td>
</tr>
<tr>
<td>OS parameters required based on the operating system chosen for SAP HANA</td>
<td>All SAP systems and patterns</td>
</tr>
<tr>
<td>Security groups created and assigned for accessing the SAP system</td>
<td>All SAP systems and patterns</td>
</tr>
<tr>
<td>SSM Session Manager to remotely access the server for administrator activities</td>
<td>All SAP systems and patterns</td>
</tr>
<tr>
<td>Time zone settings at the OS level</td>
<td>All SAP systems and patterns</td>
</tr>
</tbody>
</table>

Custom deployment configuration scripts

You can use custom shell scripts during the pre-deployment and post-deployment configuration phases. You provide the scripts stored on Amazon S3 or locally. During provisioning, Launch Wizard installs the AWSTOE application. When there are custom scripts to run, Launch Wizard creates an AWSTOE document that downloads the scripts from the location specified and then runs the scripts. The success of the custom scripts is a customer responsibility. Check the CloudWatch log streams for detailed execution logs or failure information after the scripts are deployed.

The number of configuration scripts you can use depends on the deployment model. For SAP HANA deployments, you can use one script, which runs on all of the HANA instances (both primary and worker nodes). For NetWeaver stack on SAP HANA database, the following script limits apply:

- **NetWeaver stack on SAP HANA single-instance deployment** — Because all tiers are installed on the same database instance, you can use only one script.
- **NetWeaver stack on SAP HANA distributed-instance deployment** — You can use one script per each instance tier selected, including for ABAP System Central Services (ASCS) Server and Primary Application Server (PAS), Database (DB) Server, and Additional App Servers (AAS).
- **NetWeaver stack on SAP HANA high availability deployment** — You can use one script per each instance tier selected, including for Primary Application Server (PAS), ABAP System Central Services (ASCS) Server, Database (DB) Server, Additional App Servers (AAS), and Enqueue Replication Server (ERS).

Pre-deployment configuration scripts
Pre-deployment configuration scripts run after the instances are launched and the baseline Launch Wizard configuration tasks, such as deploying Amazon CloudWatch, Amazon EC2 Systems Manager agents, and the AWS CLI, are complete. If you want to run multiple pre-deployment configuration scripts, Launch Wizard runs them in parallel on each EC2 instance in the order in which they are specified. Pre-deployment configuration scripts can be used to perform tasks such as OS hardening or deploying security and logging software. The maximum runtime for all pre-deployment configuration scripts on a single EC2 instance is 45 minutes.

Post-deployment configuration scripts

Post-deployment configuration scripts run when Launch Wizard completes configuration tasks specific to the application on all of the instances in a deployment. Before the provisioning process completes, post-configuration scripts run on all of the specified instance tiers. Launch Wizard uses SSM and AWS Lambda to trigger running post-deployment scripts on all selected SAP instances in the order in which they are specified. They can be used to perform tasks such as installing monitoring and management software, and for updating your DNS with entries for the newly deployed SAP servers and the domains joining them. The maximum runtime for all post-deployment configuration scripts on a single instance is 2 hours.

Manual cleanup activities

If you choose to delete a deployment, or a deployment fails during the deployment phase and rolls back, Launch Wizard deletes the Amazon EC2 and Amazon EBS volumes that it launches as part of the deployment. It also removes the AWSTOE application. The following resources are considered shared resources and are created without the deletion flag.

- The Amazon Elastic File System file system that is created for the SAP transport files system /usr/sap/trans.
- The Amazon Elastic File System that is created for storing SAP software and media.
- The security groups that you create.

These resources must be manually verified to ensure that they are not being used by other systems in the landscape. They must then be manually deleted from either the Amazon Elastic File System or Amazon EC2 consoles, or by using APIs.

Default Quotas

To view the default quotas for AWS Launch Wizard, see AWS Launch Wizard Endpoints and Quotas.

AWS Regions and Endpoints

To view the service endpoints for AWS Launch Wizard, see AWS Launch Wizard Endpoints and Quotas.

Get started with AWS Launch Wizard for SAP

This topic contains information to help you set up your environment and deploy AWS resources with Launch Wizard, such as:

- How to create an IAM policy and attach it to your IAM user identity
- Configuration settings to apply to your environment
- How to deploy an SAP application from the AWS Management Console
Getting started topics

- Set up for AWS Launch Wizard for SAP (p. 83)
- Deploy an SAP application with AWS Launch Wizard (p. 87)
- AWS Launch Wizard for SAP tutorials (p. 107)

Set up for AWS Launch Wizard for SAP

This section describes the prerequisites that you must verify to deploy an SAP application with AWS Launch Wizard.

Prerequisites

- General prerequisites (p. 83)
- AWS Identity and Access Management (IAM) (p. 83)

General prerequisites

The following general prerequisites must be met to deploy an application with Launch Wizard.

- You must create an Amazon VPC that consists of private subnet(s) in a minimum of two Availability Zones. The subnets must have outbound internet access. For more information on how to create and set up a VPC, see Getting Started with Amazon VPC in the Amazon VPC User Guide.
- You must create an IAM user and attach the AmazonLaunchWizard_Fullaccess policy. See the following sections (p. 83) for the steps to attach the policy to the IAM user.
- To run custom pre- and post-configuration deployment scripts, you must add the permissions listed in Add permissions to run custom pre- and post-deployment configuration scripts (p. 85) to the AmazonEC2RoleForLaunchWizard role.
- If you want to install an SAP HANA database, you must download the software from the SAP Software Download page and upload it to an Amazon S3 bucket. For steps on how to download the software and upload it to an Amazon S3 bucket, see Make SAP HANA software available for AWS Launch Wizard to deploy a HANA database (p. 109).
- Depending on the operating system version you want to use for the SAP deployment, an SAP Marketplace subscription may be required. For a complete list of supported operating system versions, see Supported operating system versions for SAP deployments (p. 69).

AWS Identity and Access Management (IAM)

Establishing the AWS Identity and Access Management (IAM) role and setting up the IAM user for permissions are typically performed by an IAM administrator for your organization. The steps are as follows:

- A one-time creation of IAM roles that Launch Wizard uses to deploy SAP systems on AWS.
- The creation of IAM users who can grant permission for Launch Wizard to deploy applications.

Launch Wizard for SAP IAM topics

- One-time creation of IAM role (p. 84)
- Create and enable IAM users to use Launch Wizard (p. 84)
- Add permissions to use AWS KMS keys (p. 84)
- Add permissions to run custom pre- and post-deployment configuration scripts (p. 85)
• Add permissions to save deployment artifacts to Amazon S3 (p. 86)

One-time creation of IAM role

On the Choose Application page of Launch Wizard, under Permissions, Launch Wizard displays the IAM role required for the Amazon EC2 instances created by Launch Wizard to access other AWS services on your behalf. When you select Next, Launch Wizard attempts to discover the IAM role in your account. If the role exists in your account, it is attached to the instance profile for the Amazon EC2 instances that Launch Wizard launches from your account. If the role does not exist, Launch Wizard attempts to create the role with the same name, AmazonEC2RoleForLaunchWizard.

The AmazonEC2RoleForLaunchWizard role is comprised of two IAM managed policies: AmazonSSMManagedInstanceCore and AmazonEC2RolePolicyForLaunchWizard. The AmazonEC2RoleForLaunchWizard role is used by the instance profile for the Amazon EC2 instances that Launch Wizard launches into your account as part of the deployment.

If you want to deploy AWS Backint Agent as a backup and restore solution for your application, you must attach a policy to the AmazonEC2RoleForLaunchWizard so that Launch Wizard can perform Backint Agent operations on your behalf. The required policy and instructions can be found in Step 2 of the Backint Agent IAM documentation. During a deployment, Launch Wizard provides the policy as well as the steps to update the role, taking user specifications into account.

After the IAM roles are created, the IAM administrator can either continue with the deployment process or optionally delegate the application deployment process to another IAM user, as described in the following section. At this point in the IAM set up process, the IAM administrator can exit the Launch Wizard service.

Create and enable IAM users to use Launch Wizard

To deploy an SAP system with Launch Wizard, the user must be assigned the AmazonLaunchWizard_Fullaccess policy. The following steps guide the IAM administrator through the process of attaching an IAM policy to an IAM user to grant that user permission to access and deploy applications from Launch Wizard.

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the left navigation pane, choose Policies.
3. Choose Users.
4. Choose the name of the user whose permissions you want to modify, and then choose the Permissions tab.
5. Choose Add permissions.
6. Choose Attach existing policies directly to user.
7. Search for the policy named AmazonLaunchWizard_Fullaccess and select the check box to the left of the policy name.
8. Choose Next: Review to see the list of policies to attach to the user.
9. Verify that the correct policy is listed, and then choose Add permissions.

**Important**
You must log in with the user associated with this IAM policy when you use Launch Wizard.

Add permissions to use AWS KMS keys

AWS Launch Wizard uses AWS default encryption keys to encrypt Amazon EBS volumes. In addition, Launch Wizard supports the use of KMS keys created and maintained in AWS KMS. You can choose to
either create new keys or use preexisting keys to encrypt your EBS volumes. You must add permissions to the KMS key policy for your key so that Launch Wizard can use your KMS key for encryption.

How to add permissions to your KMS key policy so that Launch Wizard can use your key for encryption

1. Sign in to the AWS Management Console and open the AWS Key Management Service (AWS KMS) console at https://console.aws.amazon.com/kms.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. Choose Customer managed keys in the left navigation pane.
4. Select the alias of the KMS key that you want to use to encrypt your EBS volumes.
5. Under Key users, choose Add.
6. Select the check box next to AmazonEC2RoleForLaunchWizard and the IAM user with Launch Wizard full access permissions.
7. Choose Add. Verify that AmazonEC2RoleForLaunchWizard and the IAM user with Launch Wizard full access permissions appear in the Key users list.

Add permissions to run custom pre- and post-deployment configuration scripts

To run custom pre- and post-configuration deployment scripts, you must add the following permissions to the AmazonEC2RoleForLaunchWizard role. The following steps guide you through the process of adding the required permissions for using custom scripts to the AmazonEC2RoleForLaunchWizard role.

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the left navigation pane, choose Policies.
3. Choose Users.
4. Choose the name of the user whose permissions you want to modify, and then choose the Permissions tab.
5. Choose Add inline policy.
6. Copy and paste the following policy into the JSON tab. Enter the S3 paths where your scripts are stored.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": [
            "s3:GetObject",
            "s3:GetBucketLocation"
         ],
         "Resource": [
            "arn:aws:s3:::<S3bucket1>/<S3prefix1>/<script1>",
            "arn:aws:s3:::<S3bucket2>/<S3prefix2>/<script2>",
            "arn:aws:s3:::<S3bucket1>",
            "arn:aws:s3:::<S3bucket2>"
         ]
      }]
}
```

7. Choose Review policy and enter a name for the policy.
8. Choose **Create Policy**.
9. Verify that the correct policy is listed, and then choose **Policy actions**.
10. Choose **Attach**.
11. Search for the policy named **AmazonEC2RoleForLaunchWizard** and select the check box to the left of the policy name.
12. Choose **Attach policy**.

**Important**
You must log in with the user associated with this IAM policy when you use Launch Wizard.

If the pre- or post-deployment configuration deployment scripts are expected to run additional AWS services, the permissions to use the services must also be manually added as policy to the AmazonEC2RoleForLaunchWizard.

### Add permissions to save deployment artifacts to Amazon S3

To create AWS Service Catalog products from successful deployments, which include AWS CloudFormation templates and application configuration scripts, you must provide access to an Amazon S3 location to save the generated artifacts.

The following steps guide you through adding the required permissions for saving deployment artifacts to Amazon S3 to the AmazonLaunchWizard_Fullaccess role. If the S3 bucket that you want to use to save deployment artifacts does not contain the prefix `launchwizard` in its name, you must perform the following steps to attach the required policy to the IAM user who will be performing the deployments.

1. Sign in to the AWS Management Console and open the IAM console at [https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. In the left navigation pane, choose **Users**, and choose the IAM user to which you want to grant permissions. By default, the following policy should be attached to the user: **AmazonLaunchWizard_Fullaccess**
3. Choose the **Permissions** tab.
4. Choose **Add inline policy**.
5. Copy and paste the following policy into the **JSON** tab, replacing the placeholder text. Enter the Amazon S3 path where you want to store your artifacts in the policy.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SaveLaunchWizardDeploymentArtifacts",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": ["arn:aws:s3:::${bucketName}/${bucketFolder}*$"]
        }
    ]
}
```

6. Choose **Review policy** and enter a **Name** for the policy.
7. Choose **Create Policy**.
8. Verify that the correct policy is listed for the user.
Deploy an SAP application with AWS Launch Wizard

This section contains steps for deploying an SAP application with Launch Wizard. It includes steps for various deployment paths for NetWeaver stack on SAP HANA database and SAP HANA database.

Deployment steps

• Access AWS Launch Wizard (p. 87)
• Deploy an SAP application with AWS Launch Wizard (p. 87)

Access AWS Launch Wizard

You can launch AWS Launch Wizard from the AWS Launch Wizard console located at https://console.aws.amazon.com/launchwizard.

Deploy an SAP application with AWS Launch Wizard

The following steps guide you through deploying an SAP application with AWS Launch Wizard after you have launched it from the console.

Topics

• Create a deployment (p. 87)
• Define infrastructure (p. 87)
• Application and deployment settings (p. 91)

Create a deployment

1. From the AWS Launch Wizard landing page, choose Create deployment.
2. Choose SAP.
3. Under Permissions, Launch Wizard displays the AWS Identity and Access Management (IAM) roles required for Launch Wizard to access other AWS services on your behalf. For more information about these roles and setting up IAM for Launch Wizard, see Identity and Access Management for AWS Launch Wizard (p. 214). Choose Next.

Define infrastructure

On the Define infrastructure page, define your deployment name and infrastructure settings.

1. Under the General subheading, define the following:

   • Deployment name. Enter a unique application name for your deployment.
   • Description (Optional). Provide an optional description of your deployment.
   • Enable rollback on failed deployment. By default, if a deployment fails, your provisioned resources will not be rolled back/deleted. This default configuration helps you to troubleshoot errors at the resource level as you debug deployment issues. If you want your provisioned resources to be immediately deleted if a deployment fails, select the check box.
   • Create an AWS Service Catalog product. Select the check box to package and export AWS CloudFormation templates and associated application configuration scripts to Amazon S3 and create an AWS Service Catalog product. You use these scripts to deploy and configure AWS infrastructure resources for SAP applications. If you select this option, the templates and scripts are saved to the specified Amazon S3 path. You can use the saved AWS CloudFormation templates and AWS Service Catalog products for repeated deployments of the SAP applications using
CloudFormation, AWS Service Catalog, and third-party applications integrated with AWS Service Catalog.

- **Tags (Optional).** Enter a key and value to assign metadata to your deployment. For help with tagging, see Tagging Your Amazon EC2 Resources.

2. Under the **Infrastructure – SAP landscape** subheading, configure the following infrastructure settings for your SAP landscape.

   **Configuration options**

   - Under **Configuration type**, choose whether to **Create new configuration** or **Apply saved configuration**.
   
   - Enter the following information:
     - **Configuration name.** Enter a name or short description to identify your configuration. You can save this configuration for future use.
     - **Deployment environment.** (Create new configuration, only) Choose whether to deploy into a **Production** or **Non-Production** environment.

   **Configuration details**

   If you choose to create a new configuration, enter the following information.

   - **Key pair name.** Choose an existing key pair from the dropdown list or select the link to create one. If you select **Create new key pair name**, you are directed to the Amazon EC2 console. From the Amazon EC2 console, under **Network and Security**, choose **Key Pairs**. Choose **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**.

     **Important**
     This is the only time that you can save the private key file, so download and save it in a safe place. You must specify the name of your key pair when you launch an instance, and provide the corresponding private key each time that you connect to the instance.

   Return to the Launch Wizard console, and choose the refresh button next to the **Key Pair name** dropdown list. The new key pair appears in the dropdown list. For more information about key pairs, see Amazon EC2 **Key Pairs**.

   - **Virtual Private Cloud.** Choose a VPC from the dropdown list or select the **Create VPC** link. If you select **Create VPC**, you are redirected to the VPC console to create a VPC.

   - **Availability Zone and private subnets.** You can deploy into one or two Availability Zones with up to two private subnets per Availability Zone. Different requirements are needed for different systems in the landscape. You must select two Availability Zones with a required primary and optional secondary subnet for each Availability Zone. These selections are used for each deployment according to the deployment model that you selected.

   From the dropdown lists, choose the **Availability Zones** within which you want to deploy your SAP systems and choose the private subnets. The private subnets must have outbound connectivity to the internet and to other AWS services, such as Amazon S3, AWS CloudFormation, and CloudWatch Logs. They must also be able to access the Linux repositories required for instance configuration.

   For high availability deployments, the following subnets must share a common route table:

   - subnet 1 in Availability Zone 1 and subnet 1 in Availability Zone 2
   - subnet 2 in Availability Zone 1 and subnet 2 in Availability Zone 2
To create a private subnet

- If a subnet doesn't have a route to an internet gateway, the subnet is known as a private subnet. Use the following procedure to create a private subnet. We recommend that you enable the outbound connectivity for each of your selected private subnets using a NAT gateway. To enable outbound connectivity from private subnets with public subnets, create a NAT Gateway in your chosen public subnet. Then, follow the steps in Updating Your Route Table for each of your private subnets.

- Follow the steps in Creating a Subnet in the Amazon VPC User Guide using the existing VPC that you will use in Launch Wizard.

- When you create a VPC, it includes a main route table by default. On the Route Tables page in the Amazon VPC console, you can view the main route table for a VPC by looking for Yes in the Main column. The main route table controls the routing for all subnets that are not explicitly associated with any other route table. If the main route table for your VPC has an outbound route to an internet gateway, then any subnet created using the previous step, by default, becomes a public subnet. To ensure the subnets are private, you may need to create separate route tables for your private subnets. These route tables must not contain any routes to an internet gateway. Alternatively, you can create a custom route table for your public subnet and remove the internet gateway entry from the main route table.

- Verify Connectivity. Select the check box to verify that your private subnets have outbound internet connectivity.

- Security groups. You can choose already existing security groups or Launch Wizard can create security groups that will be assigned to the EC2 instances that Launch Wizard deploys. If you choose already existing security groups, you must ensure that all of the necessary ports required to access the SAP and SAP HANA databases are open. If you choose to allow Launch Wizard to create the security groups, the security groups are created to enable the components of the cluster to communicate. Systems that are deployed with the same configuration settings can also communicate.

  If you choose an existing security group, Launch Wizard displays the security groups that will be assigned to the EC2 instances that Launch Wizard deploys. This enables the components to communicate and systems that are deployed with the same configuration settings to communicate.

- Connectivity to external systems or users. If you allowed Launch Wizard to create the security groups, then choose the Connection type and Value of the IP address or security groups required to access the SAP systems. These values can be a network segment from which the end users access the SAP systems, or downstream/upstream systems assigned a different security group in AWS or on premises.

- Proxy setting. During the launch process, the deployed Amazon EC2 instances require outbound internet access in order to:
  - Access the operating system (SUSE/RHEL) repositories.
  - Access AWS services, such as Amazon S3, CloudWatch and Systems Manager.

  An internet gateway is typically configured for outbound internet access. If you want to route internet traffic through a proxy server, enter the proxy server details. When proxy server information is provided, Launch Wizard will make the necessary environment changes to the EC2 instances during launch so that outbound internet traffic is routed through the proxy server.

- PROXY. Enter the proxy server name and port, for example http://10.0.0.140:3128 or https://10.0.0.140.3128.
• **NO_PROXY.** When a proxy server is used for outbound communication, the NO_PROXY environment variable is used to route traffic without using the proxy for the following types of traffic:
  • local communication
  • traffic to other instances within the VPC
  • traffic to other AWS services for which VPC endpoints are created

Enter a list of comma-separated values to denote hostnames, domain names, or a combination of both.

We recommend that you add all AWS service endpoints (if defined) to the NO_PROXY environment variable so that a private connection between the VPC and the service endpoint can be established in the AWS VPN. For more information on AWS service endpoints, see AWS service endpoints.

NO_PROXY is an optional parameter. If no value is entered, the following default URLs are added to the environment. Values entered for NO_PROXY at a later time are added to this list.

```shell
NO_PROXY="localhost, 127.0.0.1, 169.254.169.254, .internal, {VPC_CIDR_RANGE}"
```

**Default NO_PROXY URL details**

- **localhost**—loopback hostname
- **127.0.0.1**—loopback adapter IP
- **169.254.169.254**—EC2 metadata link-local address
- **.internal**—default DNS for the VPC
- **{VPC_CIDR_RANGE}**—CIDR block of the VPC, for example, 10.0.0/24

• **Time zone.** Choose the time zone settings to configure the timezone on the instances from the dropdown list.

• **EBS encryption.** From the dropdown list, choose whether or not to enable EBS encryption for all of the EBS volumes that are created for the SAP systems. For more information, see Amazon EBS Encryption.

• **Domain name (DNS) settings (Optional).** Select Domain Name or Route 53 from the DNS type dropdown list.
  • If you select Domain Name, you have the option to enter a domain name to maintain a Fully Qualified Domain Name (FQDN) in the `/etc/hosts` file for each instance that is launched and configured by Launch Wizard.
  • If you select Route 53, select a Route 53 hosted zone from the dropdown list. Launch Wizard will create a DNS entry for each EC2 instance launched.

  **Note**
  Before you use a Route 53 hosted zone, verify that the hosted zone is integrated with the VPC, and that the VPC DHCP options are correctly set up.

• **SAP landscape settings.** Enter the system settings for your SAP landscape.
  • **SAP System Admin User ID.** Enter the user ID for the SAP system administrator.
  • **SAP System Admin Group ID.** Enter the group ID for SAPSYS. We recommend that you replicate this number across all of your SAP systems because SAPSYS GID must be the same between systems that are part of the transport domain.
  • **SAPINST Group ID.** Enter the group ID for the SAPINST.

• **Simple Notification Service (SNS) topic ARN (Optional).** Specify an SNS topic where Launch Wizard can send notifications and alerts. For more information, see the Amazon Simple Notification Service Developer Guide. You can also choose Create SNS topic and then create
one in the Amazon SNS console. After you create an SNS topic, you can enter it in the Launch Wizard SNS field.

- After you specify the infrastructure settings, choose **Next**.

### Application and deployment settings

The following steps show the deployment paths for **NetWeaver stack on SAP HANA database** and **SAP HANA database**. Please follow the deployment steps for your deployment path.

**Topics**

- NetWeaver stack on SAP HANA database (p. 91)
- SAP HANA database (p. 102)

### NetWeaver stack on SAP HANA database

#### Application settings

On the **Configure application settings** page, enter your NetWeaver stack on SAP HANA database application settings.

1. **Application type**. Select **NetWeaver stack on SAP HANA database**. This configuration includes:
   - NetWeaver stack for a single instance, distributed instance, or multi-AZ for high availability (HA) deployment.
   - EC2 instances for the NetWeaver application tier
   - EC2 instances for SAP HANA database and optional SAP HANA database install

2. **General settings – SAP system**. Enter the settings for your SAP system.
   - **SAP System ID (SAPSID)**. An identifier for your system. The ID must be a three character, alphanumeric string.
   - **EBS Volume Type for NetWeaver application stack instances**. Choose which volume type to use for the NW application file system `/usr/sap/SAPSID` from the dropdown list.
   - **Transport Domain Controller**. Specify whether the SAP system will be the domain controller for the SAP landscape. If not, select the transport file system of the domain controller to be mounted.

3. **General Settings – SAP HANA**. Enter the settings for your SAP HANA installation.
   - **SAP HANA System ID**. Enter the identifier for your SAP HANA database. The ID must be a three character, alphanumeric string.
   - **SAP HANA Instance number**. Enter the instance number to be used for the SAP HANA installation and setup. The ID must be a two-digit number.
   - **EBS Volume Type for SAP HANA**. Select the EBS volume types to use for **SAP HANA Data**, **SAP HANA Logs**, and **SAP Others** from the dropdown lists.

   **Note**
   gp3 volumes are not supported for HANA production databases running on Xen instances (X1, X1e, R4, and R3). When you deploy HANA databases with Xen instances after choosing **Production** as the **Deployment environment** under the **Configuration options**, gp2 volumes will be used to set up SAP HANA Data and Logs on the instances you selected for the HANA database.

4. After you enter your application settings, choose **Next**.

(Use the tab for **Single instance deployment**, **Distributed instance deployment**, or **High availability deployment**, depending on your configuration)
Single instance deployment

On the Configure deployment model page, enter the deployment details for a single instance deployment.

1. **Deployment details.** Launch Wizard supports single instance deployments, distributed instance deployments, and high availability deployments. Select **Single instance deployment.**

2. **ASCS, PAS, and DB on one EC2 instance.** Enter the deployment settings for your instance.
   - **Instance details.**
     - Under **Instance sizing,** choose whether to use **AWS/Marketplace/Community images** or **Bring your own images (BYOI).**
     - **Operating System.** Select a supported operating system version for the ASCS instance. For a complete list of operating system versions supported for ASCS, see **Supported operating system versions for SAP deployments** (p. 69).
     - **AMI ID.** For BYOI, select the AMI that you want to use from the dropdown.
     - **Host name.** Enter the host name for the EC2 instance.
     - **Private IP address.** Choose whether to use an **Auto-assigned (default) IP address** or a **Custom IP address.**
       - **Auto-assign (default).** When you select this option, an IP address will be assigned for you. This is the default option.
       - **Private IP address.** When you select this option, you can enter a single IP address. Verify that this IP address is within the subnet range of the instance you are launching.
     - **Auto Recovery.** Auto recovery is an Amazon EC2 feature to increase instance availability. Select the check box to enable EC2 automatic recovery for the instance. For more information, see **Recover Your Instance** in the Amazon EC2 User Guide.
   - Under **Instance sizing,** choose whether to **Use AWS recommended resources** or **Choose instance.**
     - **Use AWS recommended resources.**
       - **Infrastructure requirements.** Choose the requirements for your recommended resources from the dropdown list.
         - **Based on CPU/Memory.** If you select this option, enter the required number of vCPU Cores and Memory. Amazon EC2 supports up to 448 logical processors. If the amount of memory required exceeds 4TB, dedicated hosts are required.
         - **SAPS (SAP Application Performance Standard).** If you select this option, enter the SAPS rating for the SAP certified instance types.
     - **Choose your instance.**
       - **Instance type.** Choose the instance type from the dropdown list.
   - **Recommended Resources.** AWS Launch Wizard displays the **Estimated monthly cost of operation** based on your instance sizing selections and the EBS volumes that will be created and attached to the launched instances. This is an estimate of AWS costs to deploy additional resources and does not include any image costs, EC2 reservations, applicable taxes, or discounts.

3. **Pre- and post-deployment configuration scripts — optional**

You can run pre- and post-deployment configuration scripts during application provisioning. For more information about how Launch Wizard accesses and deploys these scripts, see **Custom deployment configuration scripts** (p. 81).

**Pre-deployment configuration script — optional**

- **Deployment settings.** In the event of a configuration script failure or time out, select whether to ignore all failures and proceed with the deployment. If you do not select this option, when
the configuration scripts fail or time out, Launch Wizard will roll back the workload and delete all of the AWS resources created by Launch Wizard. Pre-deployment configuration scripts must finish running in 45 minutes or less.

- **Configuration script.** Choose to use a script located in Amazon S3 and enter the URL path of the script, or enter a script manually by uploading a script file. To remove the configuration script, choose **Remove script.**

**Post-deployment configuration script — optional**

- **Deployment settings.** In the event of a configuration script failure or time out, select whether to ignore all failures and proceed with the deployment. If you do not select this option, when the configuration scripts fail or time out, Launch Wizard will roll back the workload and delete all of the AWS resources created by Launch Wizard. Post-deployment configuration scripts must finish running in 2 hours or less.

- **Configuration script.** Choose to use a script located in Amazon S3 and enter the URL path of the script, or enter a script manually by uploading a script file. To remove the configuration script, choose **Remove script.**

4. After you have entered your deployment settings, choose **Next.**

(See the **SAP software installation settings** tab.)

**Distributed instance deployment**

On the **Configure SAP HANA deployment model** page, enter the deployment details for a distributed instance deployment.

1. **Deployment details.** Launch Wizard supports single instance deployments, distributed instance deployments, and high availability deployments. Select **Distributed instance deployment.**

2. **ABAP System Central Services (ASCS) Server and Primary Application Server (PAS).** Enter the deployment settings for your instance.

- **Instance details.**
  - Under **Instance sizing**, choose whether to use **AWS/Marketplace/Community images** or **Bring your own images (BYOI).**
  - **Operating System.** Select a supported operating system version for the ASCS and PAS instances. For a complete list of operating system versions supported for ASCS, see **Supported operating system versions for SAP deployments (p. 69).**
  - **AMI ID.** For BYOI, select the AMI that you want to use from the dropdown.
  - **Host name.** Enter the host name for the EC2 instances.
  - **Auto Recovery.** Auto recovery is an Amazon EC2 feature to increase instance availability. Select the check box to enable EC2 automatic recovery for the instance. For more information, see **Recover Your Instance** in the Amazon EC2 User Guide.
  - Under **Instance sizing**, choose whether to **Use AWS recommended resources** or **Choose your instance.**

- **Use AWS recommended resources.**
  - **Infrastructure requirements.** Choose the requirements for your recommended resources from the dropdown list.
    - **Based on CPU/Memory.** If you select this option, enter the required number of vCPU Cores and Memory. Amazon EC2 supports up to 448 logical processors. If the amount of memory required exceeds 4TB, dedicated hosts are required.
    - **SAPS (SAP Application Performance Standard).** If you select this option, enter the SAPS rating for the SAP certified instance types.
  - **Choose your instance.**
3. **Settings for Database (DB) Server.** Enter the deployment settings for your instance.

**Instance details.**

- **Operating System.** Select a supported operating system version for the ASCS and PAS instances. For a complete list of operating system versions supported for ASCS, see [Supported operating system versions for SAP deployments](#) (p. 69).
- **AMI ID.** For BYOI, select the AMI that you want to use from the dropdown.

**Scale up and Scale out.** Select an upgrade strategy for your system hardware to upgrade for increased data and workload.

- **Scale-up deployment.** If you choose this deployment upgrade model, enter the host name for the database.

- **Scale-out deployment.** If you choose this deployment upgrade model, enter the SAP HANA master host name, the **Number of worker nodes**, and the **Worker node hostname prefix** under Instance sizing.

**Under Instance sizing**, choose whether to **Use AWS recommended resources** or **Choose instance**.

**Use AWS recommended resources.**

- **Define requirements.** Choose the requirements for your recommended resources from the dropdown list.

- **Based on CPU/Memory.** If you select this option, enter the required number of vCPU Cores and Memory. Amazon EC2 supports up to 448 logical processors. If the amount of memory required exceeds 4TB, dedicated hosts are required.

- **SAPS (SAP Application Performance Standard).** If you select this option, enter the SAPS rating for the SAP certified instance types.

**Instance type.** Choose the instance type from the dropdown list.

**Private IP address.** Choose whether to use an **Auto-assigned (default)** IP address or a **Custom IP address**.

- **Auto-assign (default).** When you select this option, an IP address will be assigned for you. This is the default option.

- **Private IP address.** When you select this option, you can enter a single IP address. If you have selected multiple worker nodes, enter the IP addresses to assign to the instance for each selected node. Separate more than one IP address with commas. Verify that the IP addresses are within the subnet range of the instance you are launching. You must enter the same number of IP addresses as the number of nodes selected.

**Auto Recovery.** Auto recovery is an Amazon EC2 feature to increase instance availability. Select the check box to enable EC2 automatic recovery for the instance. For more information, see [Recover Your Instance](#) in the Amazon EC2 User Guide.

**Recommended Resources.** AWS Launch Wizard displays the **Estimated monthly cost of operation** based on your instance sizing selections. This is an estimate of AWS costs to deploy additional resources and does not include any applicable taxes or discounts.

4. **Settings for Additional App Servers (AAS) - optional.** Enter the deployment settings for your AAS instances.

**Instance details.**

- **Number of Additional App Servers (AAS).** Enter the number of additional application servers.
• **Naming convention for host name.** Enter the naming convention for the host name.

• **Auto Recovery.** Auto recovery is an Amazon EC2 feature to increase instance availability. Select the check box to enable EC2 automatic recovery for the instance. For more information, see *Recover Your Instance* in the Amazon EC2 User Guide.

• **Under Instance sizing, choose whether to Use AWS recommended resources or Choose your instance.**

• **Use AWS recommended resources.**
  
  • **Define requirements.** Choose the requirements for your recommended resources from the dropdown list.
  
  • **Based on CPU/Memory.** If you select this option, enter the required number of vCPU Cores and Memory. Amazon EC2 supports up to 448 logical processors. If the amount of memory required exceeds 4TB, dedicated hosts are required.
  
  • **SAPS (SAP Application Performance Standard).** If you select this option, enter the SAPS rating for the SAP certified instance types.

• **Choose your instance.**
  
  • **Instance type.** Choose the instance type from the dropdown list.
  
  • **Recommended Resources.** AWS Launch Wizard displays the Estimated monthly cost of operation based on your instance sizing selections. This is an estimate of AWS costs to deploy additional resources and does not include any applicable taxes or discounts.

5. **Pre- and post-deployment configuration scripts — optional**

You can run pre- and post-deployment configuration scripts during application provisioning. For more information about how Launch Wizard accesses and deploys these scripts, see Custom deployment configuration scripts (p. 81).

**Pre-deployment configuration script — optional**

• **Deployment settings.** In the event of a configuration script failure or time out, choose whether to proceed with the deployment. If you do not select this option, then when the configuration scripts fail or time out, Launch Wizard will roll back the workload and delete all of the AWS resources created by Launch Wizard. Pre-deployment configuration scripts must finish running in 45 minutes or less.

• **Configuration script.** You can add one or more configuration scripts depending on the number of servers you select to run scripts during the launch phase.
  
  • For each pre-deployment configuration script that you want to run, choose to use a script located in Amazon S3 and enter the URL path of the script, or upload a script file.
  
  • Select the servers to run the pre-deployment configuration scripts during the launch phase. You can choose to run pre-deployment scripts on ABAP System Central Services (ASCS) Server and Primary Application Server (PAS), Database (DB) Server, and Additional App Servers (AAS). You can add a script for each server selected.
  
  • To remove a configuration script, choose **Remove script.** To add more configuration scripts, choose **Add another script.**

**Post-deployment configuration script — optional**

• **Deployment settings.** In the event of a configuration script failure, choose whether to proceed with the deployment. If you do not select this option, then when the configuration scripts fail or time out, Launch Wizard will roll back the workload and delete all of the AWS resources created by Launch Wizard. Post-deployment configuration scripts must finish running in 2 hours or less.

• **Configuration script.** You can add one or more configuration scripts depending on the number of servers you select to run scripts during the post-deployment phase.
• For each post-deployment configuration script that you want to run, choose to use a script located in Amazon S3 and enter the URL path of the script, or upload a script file.
• Select the servers to run the post-deployment configuration scripts when an EC2 instance has been configured for use. You can choose to run the post-deployment scripts on **ABAP System Central Services (ASCS) Server** and **Primary Application Server (PAS)**, **Database (DB) Server**, and **Additional App Servers (AAS)**. You can add a script for each server selected.
• To remove a configuration script, choose **Remove script**. To add more configuration scripts, choose **Add another script**.

6. After you have entered your additional settings, choose **Next**.

(See the **SAP software installation settings** tab.)

High availability deployment

On the **Configure SAP HANA deployment model** page, enter the deployment details for the high availability deployment.

1. **Deployment details**. Launch Wizard supports single instance deployments, distributed instance deployments, and high availability deployments. Select **High availability deployment**.

2. **Settings for ABAP System Central Services (ASCS) server**. Enter the deployment settings for your instance.

   • **Instance details**.
     • Under **Image type**, choose whether to use **AWS/Marketplace/Community images** or **Bring your own images (BYOI)**.
     • **Operating System**. Select a supported operating system version for the ASCS instances. For a complete list of operating system versions supported for ASCS, see [Supported operating system versions for SAP deployments](p. 69).
     • **AMI ID**. For BYOI, select the AMI that you want to use from the dropdown list.
     • **Host name**. Enter the host name for the EC2 instance.
     • **ASCS instance number**. Enter the instance number for the SAP installation and setup, and to open up ports for security groups.

   • Under **Instance sizing**, choose whether to use **AWS recommended resources** or **Choose your instance**.

     • **Use AWS recommended resources**.
       • **Based on CPU/Memory**. If you select this option, enter the required number of vCPU Cores and Memory. Amazon EC2 supports up to 448 logical processors. If the amount of memory required exceeds 4 TB, dedicated hosts are required.
       • **SAPS (SAP Application Performance Standard)**. If you select this option, enter the SAPS rating for the SAP certified instance type.

     • **Choose your instance**.
       • **Instance type**. Choose the instance type from the dropdown list.

     • **Recommended Resources**. AWS Launch Wizard displays the **Estimated monthly cost of operation** based on your instance sizing selections. This is an estimate of AWS costs to deploy additional resources. It does not include any applicable taxes or discounts.

3. **Settings for Enqueue Replication Server (ERS)**. Enter the deployment settings for your ERS.

   • **Instance details**.
     • Under **Instance sizing**, choose whether to use **AWS/Marketplace/Community images** or **Bring your own images (BYOI)**.
     • **Operating System**. Select a supported operating system version for the ERS instance.
     • **AMI ID**. For BYOI, select the AMI that you want to use from the dropdown list.
• **Host name.** Enter the host name for the EC2 instance.
• **ERS instance number.** Enter the instance number for the SAP installation and setup, and to open up ports for security groups.

**Under Instance sizing, choose whether to Use AWS recommended resources or Choose your instance.**

• **Use AWS recommended resources.**
  • **Based on CPU/Memory.** If you select this option, enter the required number of vCPU Cores and Memory. Amazon EC2 supports up to 448 logical processors. If the amount of memory required exceeds 4TB, dedicated hosts are required.
  • **SAPS (SAP Application Performance Standard).** If you select this option, enter the SAPS rating for the SAP certified instance type.

**Choose your instance.**

• **Instance type.** Choose the instance type from the dropdown list.

**Recommended Resources.** AWS Launch Wizard displays the Estimated monthly cost of operation based on your instance sizing selections. This is an estimate of AWS costs to deploy additional resources and does not include any applicable taxes or discounts.

4. **Settings for database (DB) Server.** Enter the deployment settings for your database.

• **Under Instance sizing, choose whether to use AWS/Marketplace/Community images or Bring your own images (BYOI).**

**Instance details.**

• **Operating System.** Select a supported operating system version for the ERS instance.
• **AMI ID.** For BYOI, select the AMI that you want to use from the dropdown list.

**Primary and secondary instance details.** Enter details for both the primary and secondary instances.

• **SAP HANA host name.** Enter the host name for the SAP HANA primary and secondary instances.
• **Server site name.** Enter the primary and secondary site name for the SAP HANA system replication.

**Private IP address settings**

• **Primary instance details**

  **Private IP address.** Choose whether to use an Auto-assigned (default) IP address or a Custom IP address for your primary instance.

  • **Auto-assign (default).** When you select this option, an IP address will be assigned for you. This is the default option.

  • **Private IP address.** When you select this option, you can enter a single IP address. Verify that this IP address is within the subnet range of the instance you are launching.

• **Secondary instance details**

  **Private IP address.** Choose whether to use an Auto-assigned (default) IP address or a Custom IP address for your secondary instance.

  • **Auto-assign (default).** When you select this option, an IP address will be assigned for you. This is the default option.

  • **Private IP address.** When you select this option, you can enter a single IP address. Verify that this IP address is within the subnet range of the instance you are launching.

• **Overlay IP address.** Enter the overlay IP address to assign to the active node. The IP address should be outside of the VPC CIDR and must not be used by any other HA cluster. It is configured to always point to the active SAP HANA node.
- **Pacemaker tag name.** Enter the tag to assign to each EC2 instance. This tag is used by the pacemaker component of SLES HAE and RHEL for SAP high availability solutions and must not be used by any other EC2 instance in your account.

- Under **Instance sizing**, choose whether to **Use AWS recommended resources** or **Choose your instance**.
  - **Use AWS recommended resources.**
    - **Based on CPU/Memory.** If you select this option, enter the required number of vCPU Cores and Memory. Amazon EC2 supports up to 448 logical processors. If the amount of memory required exceeds 4 TB, dedicated hosts are required.
    - **SAPS (SAP Application Performance Standard).** If you select this option, enter the SAPS rating for the SAP certified instance type.
  - **Choose your instance.**
    - **Instance type.** Choose the instance type from the dropdown list.
  - **Recommended Resources.** AWS Launch Wizard displays the **Estimated monthly cost of operation** based on your instance sizing selections. This is an estimate of AWS costs to deploy additional resources. It does not include any applicable taxes or discounts.

5. **Primary Application Server (PAS).** Enter the deployment settings for your instance.

- **Instance details.**
  - Under **Image type**, choose whether to use **AWS/Marketplace/Community images** or **Bring your own images (BYOI)**.
  - **Operating System.** Select a supported operating system version for the ERS instance.
  - **AMI ID.** For BYOI, select the AMI that you want to use from the dropdown list.
  - **Host name.** Enter the host name for the EC2 instance.
  - **Auto Recovery.** Auto recovery is an Amazon EC2 feature to increase instance availability. Select the check box to enable Amazon EC2 automatic recovery for the instance. For more information, see Recover Your Instance in the Amazon EC2 User Guide.
  - Under **Instance sizing**, choose whether to **Use AWS recommended resources** or **Choose your instance**.
  - **Use AWS recommended resources.**
    - **Define requirements.** Choose the requirements for your recommended resources from the dropdown list.
    - **Based on CPU/Memory.** If you select this option, enter the required number of vCPU Cores and Memory. Amazon EC2 supports up to 448 logical processors. If the amount of memory required exceeds 4TB, dedicated hosts are required.
    - **SAPS (SAP Application Performance Standard).** If you select this option, enter the SAPS rating for the SAP certified instance types.
  - **Choose your instance.**
    - **Instance type.** Choose the instance type from the dropdown list.
  - **Recommended Resources.** AWS Launch Wizard displays the **Estimated monthly cost of operation** based on your instance sizing selections. This is an estimate of AWS costs to deploy additional resources. It does not include any applicable taxes or discounts.

6. **Settings for Additional App Servers (AAS) - optional.** Enter the deployment settings for your AAS instances.

- **Instance details**
  - **Number of Additional App Servers (AAS).** Enter the number of additional application servers.
  - **Naming convention for host name.** Enter the naming convention for the host name.
• **Auto Recovery.** Auto recovery is an Amazon EC2 feature to increase instance availability. Select the check box to enable Amazon EC2 automatic recovery for the instance. For more information, see Recover Your Instance in the Amazon EC2 User Guide.

• Under **Instance sizing**, choose whether to Use AWS recommended resources or Choose your instance.

  • **Use AWS recommended resources.**
    - **Infrastructure requirements.** Choose the requirements for your recommended resources from the dropdown list.
    - **Based on CPU/Memory.** If you select this option, enter the required number of vCPU Cores and Memory. Amazon EC2 supports up to 448 logical processors. If the amount of memory required exceeds 4 TB, dedicated hosts are required.
    - **SAPS (SAP Application Performance Standard).** If you select this option, enter the SAPS rating for the SAP certified instance types.

  • **Choose your instance.**
    - **Instance type.** Choose the instance type from the dropdown list.
    - **Recommended Resources.** AWS Launch Wizard displays the Estimated monthly cost of operation based on your instance sizing selections. This is an estimate of AWS costs to deploy additional resources. It does not include any applicable taxes or discounts.

7. **Pre- and post-deployment configuration scripts — optional**

You can run pre- and post-deployment configuration scripts during application provisioning. For more information about how Launch Wizard accesses and deploys these scripts, see Custom deployment configuration scripts (p. 81).

**Pre-deployment configuration script — optional**

• **Deployment settings.** Choose whether to proceed with the deployment if a configuration script fails or times out. If you do not select this option, if the configuration scripts fail or time out, Launch Wizard will roll back the workload and delete all of the AWS resources created by Launch Wizard. Pre-deployment configuration scripts must finish running in 45 minutes or less.

• **Configuration script.** You can add one or more configuration scripts depending on the number of servers that you select to run scripts during the launch phase.
  - For each pre-deployment configuration script that you want to run, choose to use a script located in Amazon S3 and enter the URL path of the script, or upload a script file.
  - Select the servers to run the pre-deployment configuration scripts during the launch phase. You can choose to run pre-deployment scripts on **Primary Application Server (PAS)**, **ABAP System Central Services (ASCS) Server**, **Database (DB) Server**, **Additional App Servers (AAS)**, and **Enqueue Replication Server (ERS)**. You can add a script for each server selected.
  - To remove a configuration script, choose Remove script. To add more configuration scripts, choose Add another script.

**Post-deployment configuration script — optional**

• **Deployment settings.** Choose whether to proceed with the deployment if a configuration script fails. If you do not select this option, if the configuration scripts fail or time out, Launch Wizard will roll back the workload and delete all of the AWS resources created by Launch Wizard. Post-deployment configuration scripts must finish running in 2 hours or less.

• **Configuration script.** You can add one or more configuration scripts depending on the number of servers that you select to run scripts during the post-deployment phase.
  - For each post-deployment configuration script that you want to run, choose to use a script located in Amazon S3 and enter the URL path of the script, or upload a script file.
• Select the servers to run the post-deployment configuration scripts when an EC2 instance has been configured for use. You can choose to run the post-deployment scripts on Primary Application Server (PAS), ABAP System Central Services (ASCS) Server, Database (DB) Server, Additional App Servers (AAS), and Enqueue Replication Server (ERS). You can add a script for each server selected.
• To remove a configuration script, choose Remove script. To add more configuration scripts, choose Add another script.

8. After you have entered all of your deployment settings, choose Next.

(See the SAP software installation settings tab.)

SAP software installation settings

On the Configure SAP application software installation page, enter the software installation details for a single instance, distributed instance, or high availability deployment.

1. **SAP application software.** Choose whether to install the SAP installation software.
   • If you choose No, choose whether to install HANA software. If you want to install HANA software, enter the S3 location for HANA media and the HANA password. Then, proceed to Step 6. If you don't want to install HANA software, proceed to Step 9.
   • If you choose Yes, provide the information listed in the following steps.

2. **Application and Version.** If you choose to install the SAP application software, select the supported application and version of the software you want to install. The following configuration fields will change based on your application software and version selections. For supported application versions, see Supported application versions for SAP deployments (p. 71).

3. **SAP application software location.** In order to install the SAP application software, Launch Wizard requires access to the relevant software and files. For instructions to provide Launch Wizard access to the application software and associated files, see Make SAP application software available for AWS Launch Wizard to deploy SAP (p. 110).
   • **SAPCAR location.** Enter the Amazon S3 path where the SAPCAR is located.
   • **Software Provisioning Manager (SWPM) location.** Enter the Amazon S3 path where the SWPM is located.
   • **Kernel software location.** Enter the Amazon S3 path where the unextracted kernel with media label is located.
   • **Installation export location.** Enter the Amazon S3 path where the installation export is located.
   • **HANA database software location.** Enter the Amazon S3 path where the SAP HANA database software is located.
   • **SAP HANA client software location.** Enter the Amazon S3 path where the SAP HANA client software is located.

4. **Installation details**

The following fields may vary according to the application selected.
   • **Schema name** and **Master password.** Enter the schema name and password to use for the HANA database.
   • **PAS instance number.** Enter the PAS instance number.
   • **ASCS virtual host name.** Enter the ASCS virtual host name used to set up high availability.
   • **ASCS virtual IP address.** Enter the ASCS virtual IP address.
   • **Enqueue Replication Server (ERS) instance number.** Enter the instance number to use for the ERS instance.
• **Enqueue Replication Server (ERS) virtual IP address.** Enter the virtual IP address used to set up high availability.

• **Enqueue Replication Server (ERS) virtual host name.** Enter the virtual host name used to set up high availability.

• **ASCS instance number.** Enter the ASCS instance number.

• **Database installation.** Choose whether or not to install the HANA database.

• **Database virtual host name.** Enter the database virtual host name used to set up high availability.

• **Software.** Select the software type that you want to install. You can install SQL or SAP software.

• **Host name.** Enter the Central Instance, ASCS, ASCS virtual IP, or Enqueue Replication Server (ERS) host name.

5. **Additional installation details.** Select the parameter name and values to use for your software installation. The following fields may vary according to the application selected.

   • **Number of batch processes.** Enter the maximum number of batch processes.

   • **Number of dialog processes.** Enter the maximum number of dialog processes.

   • **UID for SAP host agent.** Enter the UID for the SAP host agent.

   • **Create a DBA Cockpit user.** Choose whether to create a DBA Cockpit user.

6. **AWS Backint Agent.** Select the check box to install AWS Backint Agent. For more information, see AWS Backint Agent for SAP HANA.

   a. **S3 file path.** Select or enter the Amazon S3 location to store the SAP HANA backup files.

   b. **AWS KMS key ARN.** Select the ARN of the KMS key that can be used by AWS Backint Agent to encrypt the backup files. For more information, see the AWS Backint Agent for SAP documentation.

   c. **Agent version.** Select the AWS Backint Agent version you want to install.

7. **Additional preferences.**

   a. When you use AWS Backint Agent, the HANA backup files are stored in Amazon S3, which eliminates the requirement for local EBS backup volumes. If you want Launch Wizard to provision local EBS backup volumes for file-based backups that can be configured manually after deployments, select the check box.

   b. By default, a Launch Wizard deployment rolls back when the AWS Backint Agent installation fails. If you want to continue with a Launch Wizard deployment when the AWS Backint Agent installation fails, select the check box. This option does not apply to high availability deployments.

8. **IAM permissions.** To deploy an application successfully, Launch Wizard must be allowed to perform operations in other AWS services on your behalf. To do this, the Launch Wizard IAM role, AmazonEC2RoleForLaunchWizard, must have permissions attached to perform these operations, which include AWS Backint Agent operations, running pre- and post-deployment configuration scripts, and downloading the SAP installation media from Amazon S3. If the required policy is not attached to the Launch Wizard role, the Launch Wizard deployment can fail. Select the check box to verify that you have attached the required permissions before deploying.

   For steps to attach the required permissions to AmazonEC2RoleForLaunchWizard, see AWS Identity and Access Management (IAM) (p. 83) in this guide.

9. **Choose Deploy**

   (See the Review tab)
Review

- On the **Review** page, review your infrastructure, application, and deployment model settings. If you are satisfied with your selections, choose **Deploy**. If you want to change settings, choose **Previous**.
- When you choose **Deploy**, you are redirected to the **Deployments** page, where you can view the status of your deployment, and also the deployment details.

**SAP HANA database**

Application settings

On the **Configure application settings** page, enter your SAP HANA database application settings.

1. **Application type.** Select **SAP HANA database**. This configuration includes:
   - EC2 instances for an SAP HANA database
   - Optional installation of SAP HANA database software
2. **General Settings – SAP HANA.** Enter the settings for your SAP HANA database installation.
   - **SAP HANA System ID (SID).** Enter the SAP HANA system ID for your system. The HANASID must be different from SAPSID if you are configuring a single instance deployment.
   - **SAP HANA Instance number.** Enter the instance number to use for your SAP HANA system. This must be a two-digit number from 00 through 99.
   - **EBS Volume Type for SAP HANA.** Select the EBS volume types that you want to use for SAP HANA Data, SAP HANA Logs, and SAP Others from the dropdown lists.

   **Note**
   gp3 volumes are not supported for HANA production databases running on Xen instances (X1, X1e, R4, and R3). When you deploy HANA databases with Xen instances after choosing **Production** as the **Deployment environment** under the **Configuration options**, gp2 volumes will be used to set up SAP HANA Data and Logs on the instances you selected for the HANA database.

   - **SAP HANA software install.** Select whether you want to download the SAP HANA software.
     - If you select **Yes**, enter the Amazon S3 location where the SAP HANA software is located. The S3 bucket must have the prefix “launchwizard” in the bucket name to ensure that the Launch Wizard IAM role policy for EC2 has read-only access to the bucket. For steps to set up the folder structure for your S3 bucket, see **Make SAP HANA software available for AWS Launch Wizard to deploy a HANA database** (p. 109). Enter a password to use for your SAP HANA installation.

**AWS Backint Agent.** Select the check box if you want to deploy AWS Backint Agent for backup and recover along with the application. For more information about AWS Backint Agent, see **AWS Backint Agent for SAP HANA**.

- **S3 URI.** Enter the URI of the S3 bucket where you want to store your SAP HANA backup files. For example, s3://<bucket-name>.
- **S3 Encryption (AWS KMS key ARN).** Select the ARN of the KMS key that AWS Backint Agent can use to encrypt the backup files stored in your Amazon S3 bucket.
- **Agent version.** Select the version number of the agent that you want to install. If you do not enter a version number, the latest published version of the agent is installed.
- **Additional Backint preferences.**
  - If you selected to use AWS Backint agent, the agent backs up files to S3, which removes the requirement for EBS backup volumes. Select this check box to provision local EBS backup volumes for file-level backups.
• By default, Launch Wizard rolls back a deployment when the AWS Backint Agent installation fails. Select the check box if you want Launch Wizard to continue with non-HA application deployments when the Backint installation fails.

• **Verify that you have attached the required policy for Backint operations to the following role.** Select this check box after you have attached the required policy to the AmazonEC2RoleForLaunchWizard. This policy allows Launch Wizard to perform Backint Agent operations on your behalf. The policy and instructions to attach the policy to the role are provided by Launch Wizard during deployment. This information can also be found in [Step 2 of the Backint Agent IAM documentation](#).

• If you select **No**, only the AWS infrastructure is provisioned so you can manually deploy an SAP HANA database post deployment.

3. After you enter your application settings, choose **Next**.

(Use the tab for **Single instance deployment**, **Multiple instance deployment**, or **High availability deployment**, depending on your configuration)

**Single instance deployment**

On the **Configure deployment model** page, enter the deployment details for the SAP HANA database deployment.

1. **Deployment model.** Launch Wizard supports single instance deployments, multiple instance deployments, and high availability deployments. Select **Single instance deployment**.

2. **Settings for SAP HANA database on one instance**

   • **Instance details.**
     • Under **Image type**, choose whether to use AWS/Marketplace/Community images or **Bring your own images (BYOI)**.
     • **Operating System.** Select a supported operating system version for the ERS instance.
     • **AMI ID.** For BYOI, select the AMI that you want to use from the dropdown.
     • **Host name.** Enter the host name for the EC2 instance.
     • **Auto Recovery.** Auto recovery is an Amazon EC2 feature to increase instance availability. Select the check box to enable EC2 automatic recovery for the instance. For more information, see Recover Your Instance in the Amazon EC2 User Guide.

   • Under **Instance sizing**, choose **Use AWS recommended resources** or **Choose your instance**.

   • **Use AWS recommended resources.**
     • **Define requirements.** Choose the requirements for your recommended resources from the dropdown list.
     • **Based on CPU/Memory.** If you select this option, enter the required number of vCPU Cores and Memory. Amazon EC2 supports up to 448 logical processors. If the amount of memory required exceeds 4TB, dedicated hosts are required.
     • **SAPS (SAP Application Performance Standard).** If you select this option, enter the SAPS rating for the SAP certified instance types.

   • **Choose your instance.**
     • **Instance type.** Choose the instance type from the dropdown list.

   • **Recommended Resources.** Launch Wizard displays the **Estimated monthly cost of operation** based on your instance sizing selections. This is an estimate of AWS costs to deploy additional resources and does not include applicable taxes or discounts.
3. **Pre- and post-deployment configuration scripts — optional**

You can run pre- and post-deployment configuration scripts during application provisioning. For more information about how Launch Wizard accesses and deploys these scripts, see Custom deployment configuration scripts (p. 81).

**Pre-deployment configuration script — optional**

- **Deployment settings.** In the event of a configuration script failure or time out, select whether to ignore all failures and proceed with the deployment. If you do not select this option, when the configuration scripts fail or time out, Launch Wizard will roll back the workload and delete all of the AWS resources created by Launch Wizard. Pre-deployment configuration scripts must finish running in 45 minutes or less.

- **Configuration script.** Choose to use a script located in Amazon S3 and enter the URL path of the script, or enter a script manually by uploading a script file. To remove the configuration script, choose **Remove script.**

**Post-deployment configuration script — optional**

- **Deployment settings.** In the event of a configuration script failure or time out, select whether to ignore all failures and proceed with the deployment. If you do not select this option, when the configuration scripts fail or time out, Launch Wizard will roll back the workload and delete all of the AWS resources created by Launch Wizard. Post-deployment configuration scripts must finish running in 2 hours or less.

- **Configuration script.** Choose to use a script located in Amazon S3 and enter the URL path of the script, or enter a script manually by uploading a script file. To remove the configuration script, choose **Remove script.**

4. After you enter your deployment settings, choose Next.

(See the Review tab)

Multiple instance deployment

On the Configure deployment model page, enter the deployment details for the SAP HANA database deployment.

1. **Deployment model.** Launch Wizard supports single instance deployments, multiple instance deployments, and high availability deployments. Select **Multiple instance deployment.**

2. **SAP HANA on multiple EC2 instances**

- **Instance details.**
  - Under **Instance sizing,** choose whether to use **AWS/Marketplace/Community images** or **Bring your own images (BYOI).**
  - **Operating System.** Select a supported operating system version for the SAP HANA servers.
  - **AMI ID.** For BYOI, select the AMI that you want to use from the dropdown.

- Under **Instance sizing,** choose **Use AWS recommended resources** or **Choose your instance.**

- **Use AWS recommended resources.**
  - **Infrastructure requirements.** Choose the requirements for your recommended resources from the dropdown list.
  - **Based on CPU/Memory.** If you select this option, enter the required number of vCPU Cores and Memory. Amazon EC2 supports up to 448 logical processors. If the amount of memory required exceeds 4TB, dedicated hosts are required.
• **SAPS (SAP Application Performance Standard)**. If you select this option, enter the SAPS rating for the SAP certified instance types.

• **Choose your instance.**

  • **Instance type**. Choose the instance type from the dropdown list.

• **Host Name for SAP system**. Enter the host name for the EC2 instance.

• **Number of worker nodes**. Enter the number of EC2 instances to be configured as worker nodes for this SAP HANA system.

• **Worker node hostname prefix**. Enter the hostname prefix for the worker nodes.

• **Auto Recovery**. Auto recovery is an Amazon EC2 feature to increase instance availability. Select the check box to enable EC2 automatic recovery for the instance. For more information, see Recover Your Instance in the Amazon EC2 User Guide.

• **Recommended Resources**. Launch Wizard displays the **Estimated monthly cost of operation** based on your instance sizing selections. This is an estimate of AWS costs to deploy additional resources and does not include applicable taxes or discounts.

3. **Pre- and post-deployment configuration scripts — optional**

   You can run pre- and post-deployment configuration scripts during application provisioning. For more information about how Launch Wizard accesses and deploys these scripts, see Custom deployment configuration scripts (p. 81).

   **Pre-deployment configuration script — optional**

   • **Deployment settings**. In the event of a configuration script failure or time out, select whether to ignore all failures and proceed with the deployment. If you do not select this option, when the configuration scripts fail or time out, Launch Wizard will roll back the workload and delete all of the AWS resources created by Launch Wizard. Pre-deployment configuration scripts must finish running in 45 minutes or less.

   • **Configuration script**. Choose to use a script located in Amazon S3 and enter the URL path of the script, or enter a script manually by uploading a script file. To remove the configuration script, choose **Remove script**.

   **Post-deployment configuration script — optional**

   • **Deployment settings**. In the event of a configuration script failure or time out, select whether to ignore all failures and proceed with the deployment. If you do not select this option, when the configuration scripts fail or time out, Launch Wizard will roll back the workload and delete all of the AWS resources created by Launch Wizard. Post-deployment configuration scripts must finish running in 2 hours or less.

   • **Configuration script**. Choose to use a script located in Amazon S3 and enter the URL path of the script, or enter a script manually by uploading a script file. To remove the configuration script, choose **Remove script**.

4. After you enter your deployment settings, choose **Next**.

   (See **Review tab**)

**High availability deployment**

On the **Configure deployment model** page, enter the deployment details for the SAP HANA database deployment.

1. **Deployment model**. Launch Wizard supports single instance deployments, multiple instance deployments, and high availability deployments. Select **High availability deployment**.

2. **Instance details**.
• Under **Instance details**, choose whether to use **AWS/Marketplace/Community images** or **Bring your own images (BYOI)**.
  • **Operating System.** Select a supported operating system version for the SAP HANA servers.
  • **AMI ID.** For BYOI, select the AMI that you want to use from the dropdown.

• **Primary and secondary instance details.** Enter details for both the primary and secondary instances.
  • **SAP HANA host name.** Enter the host name for the SAP HANA primary and secondary instances.
  • **Server site name.** Enter the primary and secondary site name for the SAP HANA system replication.

• **Overlay IP address.** Enter the overlay IP address to assign to the active node. The IP address should be outside of the VPC CIDR and must not be used by any other HA cluster. It is configured to always point to the active SAP HANA node.

• **Pacemaker tag name.** Enter the tag to assign to each EC2 instance. This tag is used by the pacemaker component of SLES HAE and RHEL for SAP high availability solutions and must not be used by any other EC2 instance in your account.

• Under **Instance sizing**, choose **Use AWS recommended resources** or **Choose your instance**.
  • **Use AWS recommended resources.**
    • **Infrastructure requirements.** Choose the requirements for your recommended resources from the dropdown list.
      • **Based on CPU/Memory.** If you select this option, enter the required number of vCPU Cores and Memory. Amazon EC2 supports up to 448 logical processors. If the amount of memory required exceeds 4TB, dedicated hosts are required.
      • **SAPS (SAP Application Performance Standard).** If you select this option, enter the SAPS rating for the SAP certified instance types.
  • **Choose your instance.**
    • **Instance type.** Choose the instance type from the dropdown list.
  • **Recommended Resources.** Launch Wizard displays the **Estimated monthly cost of operation** based on your instance sizing selections. This is an estimate of AWS costs to deploy additional resources and does not include applicable taxes or discounts.

3. **Pre- and post-deployment configuration scripts — optional**

You can run pre- and post-deployment configuration scripts during application provisioning. For more information about how Launch Wizard accesses and deploys these scripts, see Custom deployment configuration scripts (p. 81).

**Pre-deployment configuration script — optional**

• **Deployment settings.** In the event of a configuration script failure or time out, select whether to ignore all failures and proceed with the deployment. If you do not select this option, when the configuration scripts fail or time out, Launch Wizard will roll back the workload and delete all of the AWS resources created by Launch Wizard. Pre-deployment configuration scripts must finish running in 45 minutes or less.

• **Configuration script.** Choose to use a script located in Amazon S3 and enter the URL path of the script, or enter a script manually by uploading a script file. To remove the configuration script, choose **Remove script**.
Post-deployment configuration script — optional

- **Deployment settings.** In the event of a configuration script failure or time out, select whether to ignore all failures and proceed with the deployment. If you do not select this option, when the configuration scripts fail or time out, Launch Wizard will roll back the workload and delete all of the AWS resources created by Launch Wizard. Post-deployment configuration scripts must finish running in 2 hours or less.

- **Configuration script.** Choose to use a script located in Amazon S3 and enter the URL path of the script, or enter a script manually by uploading a script file. To remove the configuration script, choose **Remove script.**

4. After you enter your deployment settings, choose Next.

(See Review tab)

**Review**

- On the Review page, review your infrastructure, application, and deployment model settings. If you are satisfied with your selections, choose **Deploy.** If you want to change settings, choose **Previous.**

- When you choose Deploy, you are redirected to the **Deployments page,** where you can view the status of your deployment, and also the deployment details.

### AWS Launch Wizard for SAP tutorials

The following tutorials can help you get started with deploying an application with AWS Launch Wizard.

**Topics**

- Deploy SAP HANA with AWS Launch Wizard (p. 107)
- Deploy SAP S/4HANA with high availability (p. 107)
- Automate a high availability configuration for SAP HANA (p. 107)

### Deploy SAP HANA with AWS Launch Wizard

For more information about how to deploy an SAP HANA database on AWS that applies AWS and SAP best practices, watch the following video.

**Deploy SAP HANA on AWS in under two hours with AWS Launch Wizard**

### Deploy SAP S/4HANA with high availability

For more information about how to automate the configuration of a high availability SAP S/4HANA application on AWS using AWS Launch Wizard, watch the following video.

**How to deploy SAP S/4HANA with high availability using AWS Launch Wizard**

### Automate a high availability configuration for SAP HANA

For more information about how to automate a high availability configuration for an SAP HANA database on AWS and then test the failover of the system with AWS Launch Wizard, watch the following video.
Demonstrating failover of SAP S/4HANA with high availability on AWS

Manage application resources with AWS Launch Wizard for SAP

After you have deployed an SAP application, you can manage and update it as follows:

- Manage deployments (p. 108)
- Delete infrastructure configuration (p. 108)

Manage deployments

1. From the left navigation pane, choose SAP.
2. Under the Deployments tab, select the check box next to the application that you want to manage, and then choose Actions. You can do the following:
   1. Manage resources on the EC2 console. You are directed to the Amazon EC2 console, where you can view and manage your SAP application resources, such as Amazon EC2, Amazon EBS, Amazon VPC, Subnets, NAT Gateways, and Elastic IPs.
   2. View resource group with Systems Manager. In the Systems Manager console, you can manage your application with built-in integrations through resource groups. Launch Wizard automatically tags your deployment with resource groups. When you access Systems Manager through Launch Wizard, the resources are automatically filtered for you based on your resource group. You can manage, patch, and maintain your applications in Systems Manager.
   3. View CloudWatch application logs. You are directed to the CloudWatch dashboard, where you can view your logs.
   4. View CloudFormation template. You are directed to the AWS CloudFormation to view the templates created for this deployment.
   5. View Service Catalog product. You are directed to the AWS Service Catalog console to view the AWS Service Catalog product that was created for this deployment.
3. To delete a deployment, select the application that you want to delete, and select Delete. You are prompted to confirm the deletion.

   Important
   When you delete a deployment, Launch Wizard attempts to delete only the AWS resources it created in your account as part of the deployment. Launch Wizard considers certain resources, such as security groups, infrastructure configuration templates created during a deployment, and EFS file systems created for a transport directory, as shared resources between multiple deployments. Shared resources are not deleted when you delete a deployment.
4. For more information about your application resources, choose the Application name. You can then view the Deployment events and Summary details for your application using the tabs at the top of the page.

Delete infrastructure configuration

1. From the left navigation pane, choose SAP.
2. Under the Saved infrastructure configurations tab, select the configuration name you want to delete, and then choose Delete. You are prompted to confirm the deletion.
Important
When you delete an infrastructure configuration, it will not be available for future deployments. Resources created from the configuration, such as VPCs, availability groups, subnets, and key pair names are not deleted.

3. For more information about an infrastructure configuration, choose the **Configuration name**.

---

**Make SAP HANA software available for AWS Launch Wizard to deploy a HANA database**

This section describes steps to download the SAP HANA software and upload it to Amazon S3 to make it available for Launch Wizard to deploy a HANA database.

**Topics**
- Download SAP HANA software (p. 109)
- Upload SAP HANA software to Amazon S3 (p. 109)

---

### Download SAP HANA software

To download the SAP HANA software, go to the **SAP Software Downloads** page and download the installation files directly to your local drive.

1. Navigate to the **SAP Software Downloads** page and log in to your account.
2. Under **Installation and Upgrades**, choose **Access Downloads»A-Z index**.
3. Choose **H** in the **Installations and Upgrades** window, and select **SAP HANA Platform Edition** from the list.
5. In the **Downloads** window, find the revision you want to download and download each file to your local drive.

**Note**
If you do not have access to the software and believe you should, contact the **SAP Global Support Customer Interaction Center**.

**Important**
Do not extract the downloaded HANA software. Instead, stage the files in your Amazon S3 bucket as is. Launch Wizard will extract the media and install the software for you.

---

### Upload SAP HANA software to Amazon S3

To upload the SAP HANA software to your Amazon S3 bucket, you must create and set up your destination bucket.

**Set up destination bucket**

1. Navigate to the Amazon S3 console at [https://console.aws.amazon.com/s3](https://console.aws.amazon.com/s3).
2. Choose **Create Bucket**.
3. In the **Create Bucket** dialog box, provide a name for your new S3 bucket with the prefix `launchwizard`. Choose the **AWS Region** where you want to create the S3 bucket, which should be a Region that is close to your location, and then choose `Create Bucket`. For detailed information
Make SAP application software available for AWS Launch Wizard to deploy SAP

This section describes steps to upload the SAP application software to Amazon S3 to make it available for Launch Wizard to deploy SAP.

AWS Launch Wizard supports the following software versions. To install a software version, you must provide the SAP software files to Launch Wizard by downloading them from the SAP Support Portal and then uploading them to Amazon S3. To access and use the files for installation, Launch Wizard requires them to be formatted according to the Amazon S3 file path syntax listed in the following table.

### Note
The software versions and CD numbers listed in the following table should be used as a reference for all of the software components required to deploy SAP, as well as for how to format the Amazon S3 path to make the software available for Launch Wizard to deploy SAP.

Launch Wizard supports NetWeaver 7.50, NetWeaver 7.52, S/4 HANA 1909, S/4 HANA 2020, and BW/4HANA 2.0. You can determine the latest CD numbers of supported applications by referring to the SAP Maintenance Planner or SAP Software Downloads.

### NetWeaver 7.52

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWPM</td>
<td>SWPM 1.0 latest version</td>
<td>SWPM10SP30_0-20009701.SAR</td>
<td>S3://Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/SWPM</td>
</tr>
<tr>
<td>SAPCAR</td>
<td>SAPCAR_1010-70006178.exe</td>
<td>S3://Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/SAPCAR</td>
<td></td>
</tr>
<tr>
<td>Exports</td>
<td>NW 7.52</td>
<td>51051806_part1.exe 51051806_part2.rar</td>
<td>S3://&lt;Your SAP software bucket&gt;/Exports/NW 7.52</td>
</tr>
</tbody>
</table>
Make SAP application software available to Launch Wizard

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel components</td>
<td>NW 7.53</td>
<td></td>
<td>igexe_12-80003187.3AR/&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Exports</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>igshelper_17-10010281.3AR/SAP &lt;Path representing NW version&gt;/Kernel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SAPEXE_700-80002573.3AR/SAP &lt;Path representing NW version&gt;/Kernel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SAPEXEDB_700-80002571.3AR/SAP &lt;Path representing NW version&gt;/Kernel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SAPHOSTAGENT49_49-20009394.3AR/SAP &lt;Path representing NW version&gt;/Kernel</td>
</tr>
<tr>
<td>SAP HANA Client</td>
<td>2.5</td>
<td>IMDB_CLIENT20_005_111-80002082.3AR/SAP &lt;Path representing NW version&gt;/HANA_Client_Software</td>
<td></td>
</tr>
</tbody>
</table>

The following HANA DB versions are supported.

**Note**
Use the latest CDs for the version.

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP HANA Database software</td>
<td>hana-20-sp04</td>
<td>51053787</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/HANA_DB_Software</td>
</tr>
<tr>
<td></td>
<td>hana-20-sp05</td>
<td>51054623</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/HANA_DB_Software</td>
</tr>
</tbody>
</table>

NetWeaver 7.50

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWPM</td>
<td>SWPM 1.0 latest version</td>
<td>SWPM10SP30_0-20009701</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/SWPM</td>
</tr>
<tr>
<td>SAPCAR</td>
<td>SAPCAR_1010-700061781</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;</td>
</tr>
</tbody>
</table>
### CD name | Versions | CD number | Amazon S3 file path
---|---|---|---
Exports | NW 7.50 | 51050829_3.ZIP | S3://<Your SAP software bucket>/<Path representing NW version>/Exports

#### Kernel components

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
</table>
| Kernel components | NW 7.53 | SWPM20SP10_3-80003424.SAR | S3://<Your SAP software bucket>/<Path representing NW version>/Kernel
| | | SAPHOSTAGENT49_49-20009394.SAR |
| SAP HANA Client | 2.5 | IMDB_CLIENT20_005_112 | S3://<Your SAP software bucket>/<Path representing NW version>/HANA_Client_Software

The following HANA DB versions are supported.

**Note**

Use the latest CDs for the version.

### CD name | Versions | CD number | Amazon S3 file path
---|---|---|---
SAP HANA Database software | hana-20-sp04 | 51054413 | S3://<Your SAP software bucket>/<Path representing NW version>/HANA_DB_Software
| hana-20-sp05 | 51054623 | S3://<Your SAP software bucket>/<Path representing NW version>/HANA_DB_Software

BW/4HANA 2021

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
</table>
| SWPM | SWPM 2.0 latest version | SWPM20SP10_3-80003424.SAR | S3://<Your SAP software bucket>
### SAPCAR

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAPCAR</td>
<td>SAPCAR_1010-70006178</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/SAPCAR</td>
</tr>
</tbody>
</table>

### Exports

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW4HANA300</td>
<td>BW4HANA300_INST_EXP through BW4HANA300_INST_EXP</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Exports</td>
</tr>
</tbody>
</table>

### Kernel components

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>igsexe</td>
<td>igsexe_0-70005417</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Kernel</td>
</tr>
<tr>
<td>igshelper</td>
<td>igshelper_17-100102</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Kernel</td>
</tr>
<tr>
<td>SAPEXE</td>
<td>SAPEXE_50-80005374</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Kernel</td>
</tr>
<tr>
<td>SAPEXEDB</td>
<td>SAPEXEDB_50-80005373</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Kernel</td>
</tr>
<tr>
<td>SAPHOSTAGENT</td>
<td>SAPHOSTAGENT54_54-80004822</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Kernel</td>
</tr>
</tbody>
</table>

### SAP HANA Client

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMDB_CLIENT</td>
<td>IMDB_CLIENT20_011</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/HANA_Client_Software</td>
</tr>
</tbody>
</table>

The following HANA DB version is supported.

**Note**

*Use the latest CDs for the version.

### SAP HANA database software

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP HANA database software</td>
<td>HANA 2.0 SPS05 Rev54 or later</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/HANA_DB_Software</td>
</tr>
</tbody>
</table>

### BW/4HANA 2.0

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWPM</td>
<td>SWPM 2.0 latest version</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path</td>
</tr>
</tbody>
</table>
Make SAP application software available to Launch Wizard

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAPCAR</td>
<td>SAPCAR_1010-70006178_SAP</td>
<td>N/A</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/SAPCAR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exports</td>
<td>BW4HANA 2.0</td>
<td>BW4HANA200_INST_EXPORT through BW4HANA200_INST_EXPORT</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Exports</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kernel components</td>
<td>NW 7.77</td>
<td>igsexe_12-80003187, igshelper_17-1001022, SAPEXE_300-80004393, SAPEXEDB_300-80004392, SAPHOSTAGENT49_49-20009394</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Kernel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAP HANA Client</td>
<td>2.5</td>
<td>IMDB_CLIENT20_005_111-80002082</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/HANA_Client_Software</td>
</tr>
</tbody>
</table>

The following HANA DB version is supported.

**Note**
*Use the latest CDs for the version.*

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP HANA database software</td>
<td>hana-20-sp05</td>
<td>51054623</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/HANA_DB_Software</td>
</tr>
</tbody>
</table>

**S/4HANA 2021**

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWPM</td>
<td>SWPM 2.0 latest version</td>
<td>SWPM20SP10_3-800034</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path</td>
</tr>
</tbody>
</table>
### SAP CAR

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAPCAR</td>
<td>SAPCAR_1010-70006178</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/SAPCAR</td>
</tr>
</tbody>
</table>

### Exports

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exports</td>
<td>S4Core 106</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Exports</td>
</tr>
</tbody>
</table>

### Kernel components

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel components</td>
<td>785 or later</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Kernel</td>
</tr>
</tbody>
</table>

### SAP HANA Client

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP HANA Client</td>
<td>2.11</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/HANA_Client_Software</td>
</tr>
</tbody>
</table>

The following HANA DB version is supported.

**Note**

*Use the latest CDs for the version.

### SAP HANA Database software

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP HANA Database software</td>
<td>HANA 2.0 SPS05 Rev54 or later</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/HANA_DB_Software</td>
</tr>
</tbody>
</table>

**S/4HANA 2020**

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWPM</td>
<td>SWPM 2.0 latest version</td>
<td></td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path</td>
</tr>
<tr>
<td>CD name</td>
<td>Versions</td>
<td>CD number</td>
<td>Amazon S3 file path</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SAPCAR</td>
<td>SAPCAR_1010-70006178</td>
<td>N/A</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/SAPCAR</td>
</tr>
<tr>
<td>Exports</td>
<td>S4Core 105</td>
<td>S4CORE105_INST_EXPORT_1.zip through S4CORE105_INST_EXPORT_24.zip</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Exports</td>
</tr>
<tr>
<td>Kernel components</td>
<td>NW 7.77</td>
<td>igsexe_0-70005417.sar</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Kernel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>igshelper_17-10010245.sar</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SAPEXE_15-70005283.SAR</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/Kernel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SAPEXEDB_15-70005282.SAR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SAPHOSTAGENT49_49-20009394.SAR</td>
<td></td>
</tr>
<tr>
<td>SAP HANA Client</td>
<td>2.5</td>
<td>IMDB_CLIENT20_005_111-80002082.SAR</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/HANA_Client_Software</td>
</tr>
</tbody>
</table>

The following HANA DB version is supported.

**Note**

*Use the latest CDs for the version.*

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP HANA Database software</td>
<td>hana-20-sp05</td>
<td>51054623</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/HANA_DB_Software</td>
</tr>
</tbody>
</table>

S/4HANA 1909

<table>
<thead>
<tr>
<th>CD name</th>
<th>Versions</th>
<th>CD number</th>
<th>Amazon S3 file path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWPM</td>
<td>SWPM 2.0 latest version</td>
<td>SWPM20SP07_0-80003424</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path</td>
</tr>
<tr>
<td>CD name</td>
<td>Versions</td>
<td>CD number</td>
<td>Amazon S3 file path</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SAPCAR</td>
<td>SAPCAR_1010-70006178</td>
<td>N/A</td>
<td>S3://&lt;Your SAP software bucket&gt;/&lt;Path representing NW version&gt;/SAPCAR</td>
</tr>
<tr>
<td>Exports</td>
<td>S4Core 104</td>
<td>S4CORE104_INST_EXPORT_1 through S4CORE104_INST_EXPORT_25</td>
<td>S3://Your SAP software bucket/Path representing NW version/Exports</td>
</tr>
<tr>
<td>Kernel components</td>
<td>NW 7.77</td>
<td>igsexe_12-80003187, igshelper_17-10010245, SAPEXE_300-80004393, SAPEXEDB_300-80004392, SAPHOSTAGENT49_49-20009394</td>
<td>S3://Your SAP software bucket/Path representing NW version/Kernel</td>
</tr>
<tr>
<td>SAP HANA Client</td>
<td>2.5</td>
<td>IMDB_CLIENT20_005_1</td>
<td>S3://Your SAP software bucket/Path representing NW version/HANA_Client_Software</td>
</tr>
</tbody>
</table>

The following HANA DB version is supported.

**Note**

*Use the latest CDs for the version.*
Repeat SAP application deployments using deployment artifacts created with AWS Launch Wizard

This section contains information about how to repeat deployments using deployment artifacts created with Launch Wizard. The artifacts include AWS Service Catalog products and AWS CloudFormation templates.

Deployment artifact topics
- How AWS Launch Wizard integration with AWS Service Catalog works (p. 118)
- Launch AWS Service Catalog products created with AWS Launch Wizard (p. 119)
- Launch AWS Service Catalog products with ServiceNow (p. 130)
- Launch AWS Service Catalog products with Jira (p. 131)
- Launch AWS Service Catalog products with Terraform (p. 131)
- Launch AWS CloudFormation templates created in Launch Wizard (p. 132)

How AWS Launch Wizard integration with AWS Service Catalog works

AWS Launch Wizard creates AWS Service Catalog products from successful deployments. The AWS Service Catalog products contain AWS CloudFormation templates and associated application configuration scripts, which are stored in Amazon S3. You can use the AWS Service Catalog products, along with integrations offered by AWS Service Catalog, with third-party products, such as ServiceNow, Jira, or Terraform. Or, you can use the AWS CloudFormation templates and application configuration scripts saved in Amazon S3 to deploy SAP applications that meet the requirements of organizational deployment and governance policies.

In addition to supporting deployments using AWS CloudFormation templates, AWS Service Catalog, and multiple deployment tools supported by AWS Service Catalog, AWS Launch Wizard creates a point-in-time snapshot of the code used to deploy and configure SAP applications at the time of the deployment. You can use the code "as is" for consistent repeated deployments, or you can use the code as a baseline and update it to meet specific application requirements.

AWS Launch Wizard creates a default Launch Wizard portfolio and products within the portfolio. An AWS Service Catalog product is created for each deployment and given a name that corresponds to the Launch Wizard deployment name.
Deploying SAP applications with Launch Wizard, AWS CloudFormation, AWS Service Catalog, and third-party applications

**Launch AWS Service Catalog products created with AWS Launch Wizard**

This section contains information to help you set up for and access AWS Service Catalog products created with AWS Launch Wizard to launch those products. It also contains information about how to create a launch constraint so that you don’t have to use your own IAM credentials to launch and manage AWS Service Catalog products.

**Topics for launching AWS Service Catalog products**
- Set up to launch AWS Service Catalog products created with AWS Launch Wizard (p. 119)
- Create a launch constraint (p. 119)
- Access AWS Service Catalog products created with AWS Launch Wizard (p. 129)
- AWS Service Catalog deployment errors (p. 129)

**Set up to launch AWS Service Catalog products created with AWS Launch Wizard**

This section provides the required steps to grant permissions to the user group. This requirement must be met to access AWS Service Catalog products created with Launch Wizard to launch those products.

**Grant AWS Service Catalog permissions to the user group**

1. Navigate to the AWS Identity and Access Management console.
2. Choose **User groups** from the left navigation pane.
3. Choose **Create group**.
4. For **User group name**, enter **Endusers**.
5. Enter **AWSServiceCatalog** in the search box to filter the policy list.
6. Select the check box next to the **AWSServiceCatalogEndUserFullAccess** policy. You can optionally choose **AWSServiceCatalogEndUserReadOnlyAccess** if you prefer to grant the user only read-only access. Choose **Create group**.
7. To add a new user to the group, in the left navigation pane, choose **Users**.
8. Choose **Add user**.
9. Enter a **User name**.
10. Select **AWS Management Console access**.
11. Choose **Next: Permissions**.
12. Choose **Add user to group**.
13. Select the check box next to the **Endusers** group, then choose **Next:Tags**.
14. Choose **Next: Review**. On the **Review** page, choose **Create user**. Download or copy the credentials, then choose **Close**.

**Create a launch constraint**

A launch constraint specifies the AWS Identity and Access Management role that AWS Service Catalog assumes when a user launches a product. It is associated with products in the portfolio. If you do not use launch constraints, you must launch and manage products using your own IAM credentials. These credentials must have permissions to use AWS CloudFormation, AWS Service Catalog, and any other AWS
services used by the products. Using a launch constraint allows you to limit the permissions of a user to the minimum required for a product.

To create a launch constraint, complete the steps in the following procedure. Perform Step 2 for each of the following listed policies.

**Create the launch role**

**AWS Service Catalog launch constraint policy 1**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "applicationinsights:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "resource-groups:List*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "route53:ChangeResourceRecordSets",
                "route53:GetChange",
                "route53:ListResourceRecordSets",
                "route53:ListHostedZones",
                "route53:ListHostedZonesByName"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": ["kms:ListKeys", "kms:ListAliases"],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": ["cloudwatch:List*", "cloudwatch:Get*", "cloudwatch:Describe*"],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": ["ec2:CreateInternetGateway", "ec2:CreateNatGateway",
```
Launch AWS Service Catalog products

```
"ec2:CreateVpc",
"ec2:CreateKeyPair",
"ec2:CreateRoute",
"ec2:CreateRouteTable",
"ec2:CreateSubnet"
],
"Resource": "*
},
{
"Effect": "Allow",
"Action": [
"ec2:AllocateAddress",
"ec2:AllocateHosts",
"ec2:AssignPrivateIpAddresses",
"ec2:AssociateAddress",
"ec2:CreateDhcpOptions",
"ec2:CreateEgressOnlyInternetGateway",
"ec2:CreateNetworkInterface",
"ec2:CreateVolume",
"ec2:CreateVpcEndpoint",
"ec2:CreateTags",
"ec2:DeleteTags",
"ec2:RunInstances",
"ec2:StartInstances",
"ec2:ModifyInstanceAttribute",
"ec2:ModifySubnetAttribute",
"ec2:ModifyVolumeAttribute",
"ec2:ModifyVpcAttribute",
"ec2:AssociateDhcpOptions",
"ec2:AssociateSubnetCidrBlock",
"ec2:AttachInternetGateway",
"ec2:AttachNetworkInterface",
"ec2:AttachVolume",
"ec2:DeleteDhcpOptions",
"ec2:DeleteInternetGateway",
"ec2:DeleteKeyPair",
"ec2:DeleteNatGateway",
"ec2:DeleteSecurityGroup",
"ec2:DeleteVolume",
"ec2:DeleteVpc",
"ec2:DetachInternetGateway",
"ec2:DetachVolume",
"ec2:DeleteSnapshot",
"ec2:AssociateRouteTable",
"ec2:AssociateVpcCidrBlock",
"ec2:DeleteNetworkAcl",
"ec2:DeleteNetworkInterface",
"ec2:DeleteNetworkInterfacePermission",
"ec2:DeleteRoute",
"ec2:DeleteRouteTable",
"ec2:DeleteSubnet",
"ec2:DetachNetworkInterface",
"ec2:DisassociateAddress",
"ec2:DisassociateVpcCidrBlock",
"ec2:GetLaunchTemplateData",
"ec2:ModifyNetworkInterfaceAttribute",
"ec2:ModifyVolume",
"ec2:AuthorizeSecurityGroupEgress",
"ec2:GetConsoleOutput",
"ec2:GetPasswordData",
"ec2:ReleaseAddress",
"ec2:ReplaceRoute",
"ec2:ReplaceRouteTableAssociation",
"ec2:RevokeSecurityGroupEgress",
"ec2:RevokeSecurityGroupIngress",
"ec2:DisassociateIamInstanceProfile",
```

121
Launch AWS Service Catalog products

Service Catalog launch constraint policy 2

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Resource": ["arn:aws:cloudformation::*::*:stack/*/*", "arn:aws:cloudformation::*::*:stack/ApplicationInsights/*"]
    },
    {
      "Effect": "Allow",
      "Action": ["ec2:StopInstances", "ec2:TerminateInstances"],
      "Resource": "*
    },
    {
      "Effect": "Allow",
      "Action": ["iam:CreateInstanceProfile", "iam:DeleteInstanceProfile", "iam:RemoveRoleFromInstanceProfile", "iam:AddRoleToInstanceProfile"],
      "Resource": ["arn:aws:iam::*:role/service-role/AmazonEC2RoleForLaunchWizard*", "arn:aws:iam::*:instance-profile/*"]
    },
    {
      "Effect": "Allow",
      "Action": ["iam:PassRole"],
      "Resource": ["arn:aws:iam::*:role/service-role/AmazonEC2RoleForLaunchWizard*", "arn:aws:iam::*:instance-profile/*"]
    }
  ]
}
```
"arn:aws:iam::*:role/service-role/AmazonLambdaRoleForLaunchWizard*",
"arn:aws:iam::*:instance-profile/**
],
"Condition": {
"StringEqualsIfExists": {
"iam:PassedToService": [
"lambda.amazonaws.com",
"ec2.amazonaws.com"
]
}
}
},
{
"Effect": "Allow",
"Action": [
"autoscaling:AttachInstances",
"autoscaling:CreateAutoScalingGroup",
"autoscaling:CreateLaunchConfiguration",
"autoscaling:DeleteAutoScalingGroup",
"autoscaling:DeleteLaunchConfiguration",
"autoscaling:UpdateAutoScalingGroup",
"logs:CreateLogStream",
"logs:DeleteLogGroup",
"logs:DeleteLogStream",
"logs:DescribeLog*",
"logs:PutLogEvents",
"resource-groups:CreateGroup",
"resource-groups:DeleteGroup",
"sns:ListSubscriptionsByTopic",
"sns:Publish",
"ssm:DeleteDocument",
"ssm:DeleteParameter*",
"ssm:DescribeDocument*",
"ssm:GetDocument",
"ssm:PutParameter"
],
"Resource": [
"arn:aws:resource-groups::*:group/**",
"arn:aws:sm:*::*",
"arn:aws:autoscaling::*:autoScalingGroup:*:autoScalingGroupName/LaunchWizard*",
"arn:aws:autoscaling::*:launchConfiguration:*:launchConfigurationName/LaunchWizard*",
"arn:aws:ssm::*:parameter/LaunchWizard*",
"arn:aws:ssm::*:document/LaunchWizard*",
"arn:aws:logs::*:log-group::*::*",
"arn:aws:logs::*:log-group:LaunchWizard*"
]
},
{
"Effect": "Allow",
"Action": "ssm:SendCommand",
"Resource": "*",
"Condition": {
"ForAllValues:StringLike": {
"aws:TagKeys": "LaunchWizard*"
}
}
},
{
"Effect": "Allow",
"Action": [
"logs:DeleteLogStream",
"logs:GetLogEvents",
"logs:PutLogEvents",
"ssm:AddTagsToResource",

Launch AWS Service Catalog products

"ssm:DescribeDocument",
"ssm:GetDocument",
"ssm:ListTagsForResource",
"ssm:RemoveTagsFromResource"
],
"Resource": [
"arn:aws:logs:*:*:log-group:*:*:*",
"arn:aws:logs:*:*:log-group:LaunchWizard***",
"arn:aws:ssm:*:*:parameter/LaunchWizard***",
"arn:aws:ssm:*:*:document/LaunchWizard***"
]
},
{
"Effect": "Allow",
"Action": [
"autoscaling:Describe*",
"cloudformation:DescribeAccountLimits",
"cloudformation:DescribeStackDriftDetectionStatus",
"cloudformation:List*",
"cloudformation:GetTemplateSummary",
"cloudformation:ValidateTemplate",
"ds:Describe*",
"ds:ListAuthorizedApplications",
"ec2:Describe*",
"ec2:Get*",
"iam:GetRole",
"iam:GetRolePolicy",
"iam:GetUser",
"iam:GetPolicyVersion",
"iam:GetPolicy",
"iam:List*",
"logs:CreateLogGroup",
"logs:GetLogDelivery",
"logs:GetLogRecord",
"logs:ListLogDeliveries",
"resource-groups:Get*",
"resource-groups:List*",
"servicequotas:GetServiceQuota",
"servicequotas:ListServiceQuotas",
"sns:ListSubscriptions",
"sns:ListTopics",
"ssm:CreateDocument",
"ssm:DescribeAutomation*",
"ssm:DescribeInstanceInformation",
"ssm:DescribeParameters",
"ssm:GetAutomationExecution",
"ssm:GetCommandInvocation",
"ssm:GetParameter*",
"ssm:GetConnectionStatus",
"ssm:ListCommand*",
"ssm:ListDocument***",
"ssm:ListInstanceAssociations",
"ssm:SendAutomationSignal",
"ssm:StartAutomationExecution",
"ssm:StopAutomationExecution",
"tag:Get**
],
"Resource": ***
},
{
"Effect": "Allow",
"Action": "logs:GetLog***",
"Resource": [
"arn:aws:logs:*:*:log-group:*:*:*",
"arn:aws:logs:*:*:log-group:LaunchWizard**"
]
Launch AWS Service Catalog products


does

effect: "allow",
action: [
  "cloudformation:list***",
  "cloudformation:describe***"
],
resource: "arn:aws:cloudformation:***:stack/launchwizard***"
},
{
  "effect": "allow",
  "action": [
    "iam:createServiceLinkedRole"
  ],
  "resource": "***",
  "condition": {
    "stringequals": {
      "iam:awsServiceName": [
        "autoscaling.amazonaws.com",
        "application-insights.amazonaws.com",
        "events.amazonaws.com"
      ]
    }
  }
},
{
  "effect": "allow",
  "action": [
    "sqs:tagQueue",
    "sqs:getQueueUrl",
    "sqs:addPermission",
    "sqs:listQueues",
    "sqs:deleteQueue",
    "sqs:getQueueAttributes",
    "sqs:listQueueTags",
    "sqs:createQueue",
    "sqs:setQueueAttributes"
  ],
  "resource": "arn:aws:sqs:***:***"
},
{
  "effect": "allow",
  "action": [
    "cloudwatch:putMetricAlarm",
    "iam:getInstanceProfile",
    "cloudwatch:deleteAlarms",
    "cloudwatch:describeAlarms"
  ],
  "resource": [
    "arn:aws:cloudwatch:***:alarm:***",
    "arn:aws:iam:***:instance-profile/***"
  ]
},
{
  "effect": "allow",
  "action": [
    "cloudformation:createStack",
    "route53:ListHostedZones",
    "ec2:createSecurityGroup",
    "ec2:AuthorizeSecurityGroupIngress",
    "elasticfilesystem:describeFileSystems",
    "elasticfilesystem:createFileSystem",
    "elasticfilesystem:createMountTarget",
    "elasticfilesystem:describeMountTargets",
    "elasticfilesystem:describeMountTargetSecurityGroups"
  ]
},
Service Catalog launch constraint policy 3

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::launchwizard*",
                "arn:aws:s3:::launchwizard/*",
                "arn:aws:s3:::aws-sap-data-provider/config.properties"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "cloudformation:TagResource",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringLike": {
                    "aws:TagKeys": "LaunchWizard*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutBucketVersioning",
                "s3:DeleteBucket",
                "lambda:CreateFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunction",
                "lambda:GetFunctionConfiguration",
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda::*:*:function:*",
                "arn:aws:s3:::launchwizard*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:CreateTable",
                "dynamodb:DescribeTable",
                "dynamodb:DeleteTable"
            ],
            "Resource": "arn:aws:dynamodb::*:*:table/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:DeleteSecret",
                "secretsmanager:TagResource"
            ],
            "Resource": "*"
        }
    ]
}
```
"secretsmanager:UntagResource",
"secretsmanager:PutResourcePolicy",
"secretsmanager:DeleteResourcePolicy",
"secretsmanager:ListSecretVersionIds",
"secretsmanager:GetSecretValue"
],
"Resource": "arn:aws:secretsmanager::*:*:secret:*"
},
{
"Effect": "Allow",
"Action": [
"secretsmanager:GetRandomPassword",
"secretsmanager:ListSecrets"
],
"Resource": "*"
},
{
"Effect": "Allow",
"Action": [
"ssm:CreateOpsMetadata"
],
"Resource": "*"
},
{
"Effect": "Allow",
"Action": "ssm:DeleteOpsMetadata",
"Resource": "arn:aws:ssm::*:*:opsmetadata/aws/ssm/LaunchWizard*"
},
{
"Effect": "Allow",
"Action": [
"sns:CreateTopic",
"sns:DeleteTopic",
"sns:Subscribe",
"sns:Unsubscribe"
],
"Resource": "arn:aws:sns::*:*"
},
{
"Effect": "Allow",
"Action": [
"fsx:UntagResource",
"fsx:TagResource",
"fsx:DeleteFileSystem",
"fsx:ListTagsForResource"
],
"Resource": "*",
"Condition": {
"StringLike": {
"aws:ResourceTag/Name": "LaunchWizard*"
}
}
},
{
"Effect": "Allow",
"Action": [
"fsx:CreateFileSystem"
],
"Resource": "*",
"Condition": {
"StringLike": {
"aws:RequestTag/Name": ["LaunchWizard*"
]
}
}
1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.

2. Perform the following substeps individually for each of the three policies previously listed.
   
   a. In the left navigation pane, choose **Policies > Create policy**.
   
   b. On the **Create policy** page, choose the **JSON** tab.
   
   c. Copy each of the previous policies and paste each into the **Policy Document** JSON text box, replacing the placeholder text.
   
   d. Choose **Next: Tags** > **Next: Review**.
   
   e. Enter a **Policy Name**.
   
   f. Choose **Create policy**.

3. In the left navigation pane, choose **Roles**, then choose **Create role**.

4. Under **Select type of trusted entity**, choose **AWS service > Service Catalog**.

5. Select the **Service Catalog** use case, then choose **Next: Permissions**.

6. Search for the three policies that you added in Step 2 and select the check boxes next to them.

7. Choose **Next: Tags**.

8. Choose **Next: Review**.

9. Enter **LaunchWizardServiceCatalogProductsLaunchRole** for the **Role name**.

10. Choose **Create role**.

### Create launch constraint

1. Navigate to the **AWS Service Catalog console**.

2. In the left navigation pane, under **Administration**, choose **Portfolios**.

3. Choose the portfolio named **Launch Wizard Service Catalog portfolio**, which is the default portfolio.

4. Under **Constraints**, choose **Create Constraints**.

5. Select the **Product** to which to apply the constraint.

6. Select **Launch** as the **Constraint type**.

7. Select the IAM role that you created in the procedure for creating a launch role.

8. Choose **Create**.
Access AWS Service Catalog products created with AWS Launch Wizard

Perform the following steps to access AWS Service Catalog products created with AWS Launch Wizard.

In the AWS Service Catalog administrator console, the **Portfolio details** page lists the portfolio settings. From this page, you can manage the products in a portfolio, grant users access to products, and apply TagOptions and constraints. You can manage products from the **Products** page.

**Access Service Catalog products as a Service Catalog Admin user**

1. Navigate to the AWS Service Catalog console.
2. In the left navigation pane, under **Administration**, choose **Portfolios**.
3. Choose the portfolio named **AWS Launch Wizard Products**, which is the default portfolio created by Launch Wizard.
4. Choose **AWS Launch Wizard products**.
5. The product created by Launch Wizard using AWS CloudFormation templates and user inputs is named **[LW Deployment Name]-[Deployment Type]**. You can create a new version by choosing **Create new version**.
6. You can associate tags or apply product-specific tags as needed.

**Access Service Catalog products as an IAM user**

1. Navigate to the AWS Service Catalog console.
2. In the left navigation pane, under **Home**, choose **Products**.
3. Search for the Launch Wizard SAP product that you saved from the Launch Wizard deployment, and select it. The product won't be visible to any user who has not been granted access to it. To grant access to the product, see **Granting Access to Users** in the AWS Service Catalog User Guide.
4. Choose **Launch product**.
5. You will be directed to the AWS Service Catalog **Launching** page, which resembles AWS CloudFormation. Most of the parameters are specified using your defaults. Enter or replace the default values as you require, including passwords and SAPSIDs.
6. After you verify the parameters, choose **Launch product** to start the creation of the AWS CloudFormation stack.

**AWS Service Catalog deployment errors**

For AWS Service Catalog deployments completed prior to February 7, 2022, perform the following steps to remove the AmazonLambdaRolePolicyForLaunchWizardSAP policy from the AmazonLambdaRoleForLaunchWizard role, and add a new inline policy. Deployments completed after February 7, 2022 do not require you to perform these steps.

1. Sign in to the AWS Management Console and open the IAM console at [https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. Choose **Roles** from the left navigation pane.
3. Search for the AmazonLambdaRoleForLaunchWizard. Select the policy to view the attached permissions.
4. Check whether the AmazonLambdaRolePolicyForLaunchWizardSAP policy is attached to this role. If it is attached, remove the policy by selecting the check box next to it, and choose **Remove**.
5. Add the following inline policy by choosing **Add permissions** > **Create inline policy**, and entering the policy in the **JSON** tab of the **Create policy** wizard.
Launch AWS Service Catalog products with ServiceNow

ServiceNow users can natively browse and provision AWS Service Catalog products created with AWS Launch Wizard by using the AWS Management Connector for ServiceNow.

Prerequisites for using ServiceNow to launch products:

- You must create a deployment using Launch Wizard by choosing the Create an AWS Service Catalog product option in the infrastructure settings in Launch Wizard. For more information, see Define infrastructure (p. 87).
- You must install the AWS Service Catalog Connector for ServiceNow. For details about how to install the Connector, see AWS Service Management Connector for ServiceNow.
- You must complete the set up steps to launch AWS Service Catalog products (p. 119).
- You must create a launch constraint (p. 119).

For more information about how to integrate AWS products into your ServiceNow Portal using the AWS Service Catalog Connector, watch the following video.
Integrate AWS Products into Your ServiceNow Portal via the AWS Service Management Connector

Launch AWS Service Catalog products with Jira

AWS Service Catalog products created with AWS Launch Wizard can be integrated with Jira workflows. You can use the AWS Service Catalog Connector for Jira to natively provision and operate AWS Service Catalog products created with Launch Wizard by using Atlassian’s Jira Service Management. This workflow simplifies product request actions for Jira Service Management users and provides Jira Service Management governance and oversight over AWS products.

To use Jira to launch products, you must follow these prerequisites:

- Create a deployment using Launch Wizard by choosing the Create an AWS Service Catalog product option in the infrastructure settings in Launch Wizard. For more information, see Define infrastructure (p. 87).
- Install the AWS Service Catalog Connector for Jira. For information about how to install the Connector, see AWS Service Management Connector for ServiceNow.
- Complete the set up steps to launch AWS Service Catalog products (p. 119).
- Complete the steps to create a launch constraint (p. 119).

For more information about how to integrate AWS products into your Jira Service Management portal using the AWS Service Catalog Connector, watch the following video.

Integrate AWS Products into Your Jira Service Management Portal

Launch AWS Service Catalog products with Terraform

The official HashiCorp AWS provider supports AWS Service Catalog resources. You can launch products created with Launch Wizard and saved to AWS Service Catalog using Terraform. Or, you can integrate the products with their existing Terraform workflows. Administrators can create AWS Service Catalog portfolios and add Launch Wizard products to them using Terraform.

Prerequisites for using Terraform to launch products:

- You must create a deployment using Launch Wizard by choosing the Create an AWS Service Catalog product option in the infrastructure settings in Launch Wizard. For more information, see Define infrastructure (p. 87).
- The Terraform user that authenticates the AWS account must have access to the AWS Service Catalog products. For more information, see AWS Provider in the Terraform documentation.
- The IAM user that authenticates the AWS account must have permissions to use the AWS Service Catalog products created by Launch Wizard. For steps to grant access to users, see Granting Access to Users in the AWS Service Catalog User Guide.

The Terraform resource named aws_servicecatalog_product is used to launch the AWS Service Catalog product created with Launch Wizard.

Example Terraform script

The following example Terraform script launches a single node HANA database instance with a single node HANA product (prod-abc1234546) created with Launch Wizard using the product version ID (pa-xyz12345). In this example, the hostname for HANA and the SID for HANA DB are passed to override the defaults, and the remaining parameters are set to the defaults in the AWS Service Catalog product.

```
terraform {
  required_providers {
    aws = {
      ..
    }
  }
```
Launch AWS CloudFormation templates created in Launch Wizard

You can launch AWS CloudFormation stacks from the AWS CloudFormation templates that you saved from your successful Launch Wizard deployments. Perform the following steps to find and launch your AWS CloudFormation templates created with Launch Wizard.

To create a launch constraint, complete the steps in the following procedure. Perform Step 2 for each of the following listed policies.

Attach required policies to IAM user

Service Catalog launch constraint policy 1

```json
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Action": "applicationinsights:*",
      "Resource": "**"
   },
   {
      "Effect": "Allow",
      "Action": "resource-groups:List***",
      "Resource": "***"
   },
   {
      "Effect": "Allow",
      "Action": "**",
      "Resource": "***"
   }
   
   
   
```
"route53:ChangeResourceRecordSets",
"route53:GetChange",
"route53:ListResourceRecordSets",
"route53:ListHostedZones",
"route53:ListHostedZonesByName"
],
"Resource": "*
},
{
"Effect": "Allow",
"Action": [
"s3:ListAllMyBuckets",
"s3:ListBucket",
"s3:GetBucketLocation"
],
"Resource": "*
},
{
"Effect": "Allow",
"Action": [
"kms:ListKeys",
"kms:ListAliases"
],
"Resource": "*
},
{
"Effect": "Allow",
"Action": [
"cloudwatch:List*",
"cloudwatch:Get*",
"cloudwatch:Describe*
]
},
"Resource": "*
},
{
"Effect": "Allow",
"Action": [
"ec2:CreateInternetGateway",
"ec2:CreateNatGateway",
"ec2:CreateVpc",
"ec2:CreateKeyPair",
"ec2:CreateRoute",
"ec2:CreateRouteTable",
"ec2:CreateSubnet"
],
"Resource": "*
},
{
"Effect": "Allow",
"Action": [
"ec2:AllocateAddress",
"ec2:AllocateHosts",
"ec2:AssignPrivateIpAddresses",
"ec2:AssociateAddress",
"ec2:CreateDhcpOptions",
"ec2:CreateEgressOnlyInternetGateway",
"ec2:CreateNetworkInterface",
"ec2:CreateVolume",
"ec2:CreateVpcEndpoint",
"ec2:CreateTags",
"ec2:DeleteTags",
"ec2:RunInstances",
"ec2:StartInstances",
"ec2:ModifyInstanceAttribute",
"ec2:ModifySubnetAttribute",
"ec2:ModifyVolumeAttribute",
"ec2:ModifyVolumeAttribute"
Launch AWS CloudFormation templates created in Launch Wizard

Service Catalog launch constraint policy 2

```json
{
  "Version": "2012-10-17",
  "Statement": [
    
  ]
}
```
"cloudformation:DescribeStack",
"cloudformation:Get",
"cloudformation:ListStacks",
"cloudformation:SignalResource",
"cloudformation:DeleteStack"
],
"Resource": [ 
  "arn:aws:cloudformation:*:*:stack/**",
  "arn:aws:cloudformation:*:*:stack/ApplicationInsights/**"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:StopInstances",
    "ec2:TerminateInstances"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateInstanceProfile",
    "iam:DeleteInstanceProfile",
    "iam:RemoveRoleFromInstanceProfile",
    "iam:AddRoleToInstanceProfile"
  ],
  "Resource": [ 
    "arn:aws:iam::*:role/service-role/AmazonEC2RoleForLaunchWizard*",
    "arn:aws:iam::*:instance-profile/**"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iam:PassRole"
  ],
  "Resource": [ 
    "arn:aws:iam::*:role/service-role/AmazonEC2RoleForLaunchWizard*",
    "arn:aws:iam::*:role/service-role/AmazonLambdaRoleForLaunchWizard*",
    "arn:aws:iam::*:instance-profile/**"
  ],
  "Condition": { 
    "StringEqualsIfExists": { 
      "iam:PassedToService": [ 
        "lambda.amazonaws.com",
        "ec2.amazonaws.com"
      ]
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "autoscaling:AttachInstances",
    "autoscaling:CreateAutoScalingGroup",
    "autoscaling:CreateLaunchConfiguration",
    "autoscaling:DeleteAutoScalingGroup",
    "autoscaling:DeleteLaunchConfiguration",
    "autoscaling:UpdateAutoScalingGroup",
    "logs:CreateLogStream",
    "logs:DeleteLogStream",
    "logs:DeleteLogGroup",
    "logs:DescribeLog",
    "logs:PutLogEvents",
    "resource-groups:CreateGroup",
    "resource-groups:CreateGroup"
  ]
}
Launch AWS CloudFormation templates created in Launch Wizard

"resource-groups:DeleteGroup",
"sns:ListSubscriptionsByTopic",
"sns:Publish",
"ssm:DeleteDocument",
"ssm:DeleteParameter",
"ssm:DescribeDocument",
"ssm:GetDocument",
"ssm:PutParameter"

"Resource": [
  "arn:aws:resource-groups::*:group/*",
  "arn:aws:sns::*:*",
  "arn:aws:autoscaling::*:autoScalingGroup::*:autoScalingGroupName/LaunchWizard*",
  "arn:aws:autoscaling::*:launchConfiguration::*:launchConfigurationName/LaunchWizard*",
  "arn:aws:ssm::*:parameter/LaunchWizard*",
  "arn:aws:ssm::*:document/LaunchWizard*",
  "arn:aws:logs::*:log-group:*:*:*",
  "arn:aws:logs::*:log-group:LaunchWizard*"
]

"Effect": "Allow",
"Condition": {
  "ForAllValues:StringLike": {
    "aws:TagKeys": "LaunchWizard*"
  }
}

"Action": [
  "ssm:SendCommand",
  "ssm:AddTagsToResource",
  "ssm:DescribeDocument",
  "ssm:GetDocument",
  "ssm:ListTagsForResource",
  "ssm:RemoveTagsFromResource"
]

"Resource": [
  "arn:aws:logs::*:log-group:*:*:*",
  "arn:aws:logs::*:log-group:LaunchWizard*",
  "arn:aws:ssm::*:parameter/LaunchWizard*",
  "arn:aws:ssm::*:document/LaunchWizard*"
]

"Effect": "Allow",
"Action": [
  "autoscaling:Describe*",
  "cloudformation:DescribeAccountLimits",
  "cloudformation:DescribeStackDriftDetectionStatus",
  "cloudformation:List*",
  "cloudformation:GetTemplateSummary",
  "cloudformation:ValidateTemplate",
  "ds:Describe*",
  "ds:ListAuthorizedApplications",
  "ec2:Describe*",
  "ec2:Get*",
  "iam:GetRole",
  "iam:GetRolePolicy",
]
"iam:GetUser",
"iam:GetPolicyVersion",
"iam:GetPolicy",
"iam:List*",
"logs:CreateLogGroup",
"logs:GetLogDelivery",
"logs:GetLogRecord",
"logs:ListLogDeliveries",
"resource-groups:Get*",
"resource-groups:List*",
"servicequotas:GetServiceQuota",
"servicequotas:ListServiceQuotas",
"sns:ListSubscriptions",
"sns:ListTopics",
"ssm:CreateDocument",
"ssm:DescribeAutomation*",
"ssm:DescribeInstanceInformation",
"ssm:GetAutomationExecution",
"ssm:GetCommandInvocation",
"ssm:GetParameter*",
"ssm:GetConnectionStatus",
"ssm:ListCommand*",
"ssm:ListDocument*",
"ssm:ListInstanceAssociations",
"ssm:SendAutomationSignal",
"ssm:StartAutomationExecution",
"ssm:StopAutomationExecution",
"tag:Get**"
],
"Resource": "*
},
{
"Effect": "Allow",
"Action": "logs:GetLog*",
"Resource": [
   "arn:aws:logs:*:*:log-group:*:*:*",
   "arn:aws:logs:*:*:log-group:LaunchWizard**
]
},
{
"Effect": "Allow",
"Action": [
   "cloudformation:List*",
   "cloudformation:Describe*"
],
"Resource": "arn:aws:cloudformation:*:*:stack/LaunchWizard*/"}
},
{
"Effect": "Allow",
"Action": [
   "iam:CreateServiceLinkedRole"
],
"Resource": "*
",
"Condition": {
   "StringEquals": {
      "iam:AWSServiceName": [
   "autoscaling.amazonaws.com",
   "application-insights.amazonaws.com",
   "events.amazonaws.com"
   ]
   }
   }
}
}
{
"Effect": "Allow",
}
"Action": [
    "sqs:TagQueue",
    "sqs:GetQueueUrl",
    "sqs:AddPermission",
    "sqs:ListQueues",
    "sqs:DeleteQueue",
    "sqs:GetQueueAttributes",
    "sqs:ListQueueTags",
    "sqs:CreateQueue",
    "sqs:SetQueueAttributes"
],
"Resource": "arn:aws:sqs:*:*:*"
},
{
"Effect": "Allow",
"Action": [
    "cloudwatch:PutMetricAlarm",
    "iam:GetInstanceProfile",
    "cloudwatch:DeleteAlarms",
    "cloudwatch:DescribeAlarms"
],
"Resource": [
    "arn:aws:cloudwatch:*:*:alarm:*",
    "arn:aws:iam::*:instance-profile/*"
]
},
{
"Effect": "Allow",
"Action": [
    "cloudformation:CreateStack",
    "route53:ListHostedZones",
    "ec2:CreateSecurityGroup",
    "ec2:AuthorizeSecurityGroupIngress",
    "elasticfilesystem:DescribeFileSystems",
    "elasticfilesystem:CreateFileSystem",
    "elasticfilesystem:CreateMountTarget",
    "elasticfilesystem:DescribeMountTargets",
    "elasticfilesystem:DescribeMountTargetSecurityGroups"
],
"Resource": "*"
}
}
}

**Service Catalog launch constraint policy 3**

{
"Version": "2012-10-17",
"Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::launchwizard*",
        "arn:aws:s3:::launchwizard/*/config.properties",
        "arn:aws:s3:::aws-sap-data-provider/config.properties"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "cloudformation:TagResource",
      "Resource": "arn:aws:cloudformation:*:*:*"
    }
  ]
}
"Resource": "*",
"Condition": {
  "ForAllValues:StringLike": {
    "aws:TagKeys": "LaunchWizard*"
  }
},
"Effect": "Allow",
"Action": [
  "s3:CreateBucket",
  "s3:PutBucketVersioning",
  "s3:DeleteBucket",
  "lambda:CreateFunction",
  "lambda:DeleteFunction",
  "lambda:GetFunction",
  "lambda:GetFunctionConfiguration",
  "lambda:InvokeFunction"
],
"Resource": [
  "arn:aws:lambda:*:*:function:*",
  "arn:aws:s3:::launchwizard*"
],
"Effect": "Allow",
"Action": [
  "dynamodb:CreateTable",
  "dynamodb:DescribeTable",
  "dynamodb:DeleteTable"
],
"Resource": "arn:aws:dynamodb:*:*:table/*"
],
"Effect": "Allow",
"Action": [
  "secretsmanager:CreateSecret",
  "secretsmanager:DeleteSecret",
  "secretsmanager:TagResource",
  "secretsmanager:UntagResource",
  "secretsmanager:PutResourcePolicy",
  "secretsmanager:DeleteResourcePolicy",
  "secretsmanager:ListSecretVersionIds",
  "secretsmanager:GetSecretValue"
],
"Resource": "arn:aws:secretsmanager:*:*:secret:*"
],
"Effect": "Allow",
"Action": [
  "secretsmanager:GetRandomPassword",
  "secretsmanager:ListSecrets"
],
"Resource": "*
],
"Effect": "Allow",
"Action": [
  "ssm:CreateOpsMetadata"
],
"Resource": "*
],
"Effect": "Allow",
"Action": "ssm:DeleteOpsMetadata",
"Resource": "arn:aws:ssm:*:*:opsmetadata/aws/ssm/LaunchWizard*"
1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. Perform the following substeps for each of the three policies listed above.

   a. In the left navigation pane, choose Policies > Create policy.
   b. On the Create policy page, choose the JSON tab.
   c. Copy each policy above and paste it into the Policy Document JSON text field, replacing the placeholder text (perform these substeps individually for each of the three policies listed above).
   d. Choose Next: Tags > Next: Review.
   e. Enter a Policy Name.
   f. Choose Create policy.

3. Attach the three policies you just created to the IAM user you use to launch AWS CloudFormation templates.

Find and launch your templates

1. Navigate to the Amazon S3 console.
2. Locate the name of the location within the Amazon S3 bucket that you specified when you defined the infrastructure for your Launch Wizard deployment (p. 87).
3. Under the folder that you specified, locate and choose a new folder named `<LaunchWizardDeploymentName>-<TimeStamp>`. This is the folder to which the Launch Wizard service copies the AWS CloudFormation templates and deployment artifacts.
4. After you choose the new folder, you will see an sap/ folder and a JSON file named `<LaunchWizardDeploymentName>-<DeploymentType>-template.json`. This is the root AWS CloudFormation template file. Select the check box next to this file and choose Copy URL.
5. Navigate to the AWS CloudFormation console to create a stack with the URL that you copied.

For more information about CloudFormation templates, see Working with AWS CloudFormation templates in the AWS CloudFormation User Guide.

Scale SAP applications with AWS Launch Wizard for SAP after initial deployment

You can scale an SAP application horizontally to meet increased performance requirements, depending on the initial SAP product and deployment pattern. This section describes how you can add additional nodes to a preexisting SAP application deployed with AWS Launch Wizard for SAP. It includes the prerequisites required to use this feature, and manual activities that you must perform after you add nodes.

Topics for scaling SAP applications:

- Shared responsibility model (p. 142)
- Prerequisites for creating an AMI (p. 142)
- Create an image for scaling SAP deployments (p. 142)
- Supported scenarios for adding or removing nodes with Launch Wizard for SAP (p. 143)
- Pre- and post-deployment configuration scripts (p. 147)
Shared responsibility model

According to the AWS Shared Responsibility Model, scaling an SAP application that was deployed with AWS Launch Wizard requires prescribed manual activities that you must complete before adding or removing nodes. These manual activities may vary depending on the scenario, for example, adding an application or database node. The activities may also vary based on the source deployment architecture, for example, multi-node or single-node. For example, you assume the responsibility of creating an Amazon Machine Image (AMI) of an existing application server or HANA worker node, upon which this feature depends. It is your responsibility to ensure that the provided image is bootable in the VPC and subnet, and that Launch Wizard can access the instance through AWS Systems Manager. AWS relieves the operational burden by using the image to provision the infrastructure as a new instance, and by installing the application, configuring it, or both.

Prerequisites for creating an AMI

Before you create an AMI to attach additional nodes, make sure that you:

- Have a successful initial deployment.
- Create an AMI from the deployment to which you are adding the node. If you want to add nodes to multiple deployments, you must create an AMI (application/HANA worker) for each deployment.

When an application is provisioned with Launch Wizard, multiple packages are installed, and operating system parameters are adjusted to make the operating system compliant and ready to install SAP. Periodically, these packages and parameters get updated and change over time. In addition, there can be changes to the disk layout as the application grows. This configuration drift is not tracked by Launch Wizard, and therefore Launch Wizard is not able to record and replay it. To mitigate the challenges presented by configuration drift, you must create and provide the latest image (AMI) of an existing server, including all of its volumes, which Launch Wizard uses to create the additional server or HANA worker node. This process allows Launch Wizard to create new nodes that are similar to the existing nodes in terms of operating system packages and versions, and storage.

Create an image for scaling SAP deployments

The following best practices for creating an image to add an additional application node to a previously deployed SAP application using Launch Wizard are general guidelines only. They do not represent a complete solution. These recommendations are offered as considerations that may not be appropriate or sufficient for your environment, depending on the activities that you performed on these instances after the Launch Wizard deployment.

General recommendations:

- Do not share images with untrusted accounts.
- Do not make public images that contain private or sensitive data.
- Apply all of the latest available operating system security patches.

Before you create an image:

- Temporarily adjust the SAP or HANA instance profile parameter Autostart to 0. Revert to the original value after the image is created.
- Temporarily disable any third-party applications from starting on boot.
- Keep the file systems and volumes intact, and ensure that the image is bootable if the volumes are not attached. If the volumes are not attached, the /etc/fstab nofail setting must be enabled.
Supported scenarios for adding or removing nodes with Launch Wizard for SAP

Add or remove an additional server or node by using an Amazon EC2 Systems Manager (SSM) document that is created in your account during runtime. The following scenarios are supported for the source deployment.

<table>
<thead>
<tr>
<th>AWS Launch Wizard for SAP application type</th>
<th>Deployment architecture</th>
<th>Supported scenario</th>
<th>AMI required</th>
</tr>
</thead>
<tbody>
<tr>
<td>HANA</td>
<td>Multi-node</td>
<td>Add a HANA worker node</td>
<td>Worker node of source deployment</td>
</tr>
<tr>
<td>NetWeaver on HANA</td>
<td>Multi-node</td>
<td>Add an application server node</td>
<td>PAS/AAS of source deployment</td>
</tr>
</tbody>
</table>

Supported scenarios:
- Add HANA worker node to existing HANA scale-out installation (p. 143)
- Add additional application server (AAS) to existing NetWeaver distributed installation (p. 145)

Add HANA worker node to existing HANA scale-out installation

Prerequisites

For prerequisites, see Prerequisites for creating an AMI (p. 142).

Assumptions

Before you proceed with this procedure, consider the following assumptions:

- A worker node will be created using the same instance type as an existing worker node. Verify that all of the worker nodes are running on the same Amazon EC2 instance type.
- All HANA nodes and respective services are up and running.
- There are no upgrades or patching in progress.
- No maintenance activities, such as backups, are in progress.

Workflow for adding a HANA worker node
When you add an additional HANA worker node to your existing HANA scale-out installation, Launch Wizard for SAP performs the following:

1. An instance is created using the provided AMI.
2. The hostname is updated when the instance boots.
3. `/etc/hosts` is updated on the master node, and then the host file is synced to the newly created node.
4. All abandoned services and processes are cleaned up.
5. If pre-deployment configuration scripts are provided, they are run.
6. `/usr/sap`, `/hana/data`, and `/hana/log` folders are cleaned up.
7. `saphostagent` is set up.
8. The HANA worker node is set up using `add_hosts`.
9. If post-deployment configuration scripts are provided, they are run.

**Console steps for adding a HANA worker node to an existing scale-out installation**

Perform the following steps in the Launch Wizard for SAP console to add a HANA worker node:

1. Navigate to the Launch Wizard for SAP Deployments page.
2. Select the check box next to the deployment to which you want to add a new node. From the Action menu, select "Deploy additional components > HANA worker node". You will be taken to the Configure settings page.
3. Under **Settings for additional worker node**, specify the following parameters:
   - **HANA worker node AMI** — select the AMI to provision the worker node. You must use the most recent version of the AMI generated from the source deployment to which the worker node is being added.
   - **Hostname of worker node** — Enter the hostname of the Amazon EC2 instance on which the SAP system will be deployed.
   - **Private IP address** (optional) — Select the private IP address to assign to the new instance. If you do not provide an IP address, Launch Wizard assigns one for you.
4. Under **Pre-deployment configuration script**, optionally specify the following parameters:
   - **Deployment settings** — Select the check box to ignore all deployment failures and proceed with a deployment.
   - **Configuration script** — Add one or more configuration scripts, depending on the number of servers included in the deployment. The scripts run in the order they are added. You can view detailed execution logs or failure information in the Amazon CloudWatch logs after a deployment is complete.
5. Under **Post-deployment configuration script**, optionally specify the following parameters:
   - **Deployment settings** — Select the check box to ignore all deployment failures and proceed with a deployment.
   - **Configuration script** — Add one or more configuration scripts, depending on the number of servers included in the deployment. The scripts run in the order they are added. You can view detailed execution logs or failure information in the Amazon CloudWatch logs after a deployment is complete.
6. Choose **Next** when the preceding parameters are specified, and the **Review infrastructure configuration** page opens.
7. Under **Infrastructure configuration**, specify the following parameters:
   - **Key pair name** — Select a key pair to securely connect to your instance.
Supported scenarios

- **Virtual Private Cloud (VPC)** — the VPC in which the domain controllers will be deployed is the same as for the original deployment.
- **Private subnet** — The private subnet is determined by the subnet specified in your original deployment.
- **Security group assigned to database servers** — Select a security group that is currently assigned to a database node.
- **HANA password** — enter a password for the SAP HANA installation.

8. Choose **Next** when you are satisfied with your infrastructure configuration selections. You will be taken to the **Review** page.

9. On the **Review** page, verify your settings and configuration. Choose **Deploy** if you are satisfied with your selections. To edit your selections, choose **Edit** or **Previous**. It can take up to 10 minutes to create your new node.

**Manual activities required**

The following manual activities are required to successfully add a HANA worker node to an existing scale-out installation.

- Host entries are updated only on the HANA master and newly added nodes. Refresh `/etc/hosts` entries from the HANA master node on all of the other existing nodes.
- When the automation workflow runs, a new HANA worker node is attached to the existing HANA deployment. The node is ready to be used. A HANA table redistribution plan must be determined and performed. For more information about how to redistribute the tables to the new nodes, see *Redistributing Tables in a Scaleout SAP HANA System* in the SAP documentation.
- The newly added worker node is not set up in the same placement group. Attach the new worker node to the placement group and restart all of the HANA nodes for the placement groups to take effect.

**Delete a HANA worker node from an existing scale-out installation**

The process of deleting a HANA worker node from an existing scale-out installation is partially automated. Before you delete a worker node, you must redistribute the data for a multi-database container (MDC) before deleting the node. For more information about how to redistribute the tables to the new nodes, see *Redistributing Tables in a Scaleout SAP HANA System* in the SAP documentation.

**Note**

You can only delete a node that was created with the add or remove node feature using Launch Wizard for SAP.

**Add additional application server (AAS) to existing NetWeaver distributed installation**

**Prerequisites**

For prerequisites, see *Prerequisites for creating an AMI* (p. 142).

**Assumptions**

The following assumptions should be considered before proceeding with this procedure.

- All servers of the SAP application to which the node is being added are running.
- There are no upgrades or patching in progress.

**Workflow for adding an additional application server (AAS) node**
When you add an additional SAP server to your existing NetWeaver distributed installation, depending on whether you provide a PAS or AAS image, the file systems and volumes are adjusted to the requirements of the additional application server. Launch Wizard for SAP performs the following:

1. An instance is created using the provided AMI.
2. The hostname is updated when the instance boots.
3. All abandoned services and processes are cleaned up.
4. `/etc/hosts` is updated on the master node, and then the hosts file is synced to the new node.
5. If the provided AMI is from the Primary Application Server (PAS) and Launch Wizard detects the `/samnt` volume in `/etc/fstab/`, the volume is unmounted and removed. `/sapmnt` is mounted as an NFS from the PAS.
6. If pre-deployment configuration scripts are provided, they are run.
7. `.env` files are updated and a new instance profile file is created.
8. Services are started.
9. The instance is started.
10. If post-deployment configuration scripts are provided, they are run.

### Console steps for adding an AAS node to an existing NetWeaver distributed installation

Perform the following steps in the Launch Wizard for SAP console to add an AAS node:

1. Navigate to the Launch Wizard for SAP **Deployments** page.
2. Select the check box next to the deployment to which you want to add a new node. From the **Action** menu, select **Deploy additional components** > **Additional application server (AAS)**. The **Configure settings** page opens.
3. Under **Settings for additional worker node**, specify the following parameters:
   - **Additional application server (AAS) AMI** — Select the AMI to provision the additional application server. You must use the most recent version of the AMI created from the source deployment.
   - **Hostname of additional application server** — Enter the hostname of the Amazon EC2 instance on which the SAP system will be deployed.
   - **Instance number of AAS** — Enter the instance number of the additional application server.
   - **Private IP address** — Select the private IP address to assign to the new instance.
4. Under **Pre-deployment configuration script**, optionally specify the following parameters:
   - **Deployment settings** — Select the check box to ignore all deployment failures and proceed with a deployment.
   - **Configuration script** — Add one or more configuration scripts, depending on the number of servers included in the deployment. The scripts run in the order they are added. You can view detailed execution logs or failure information in the Amazon CloudWatch logs after a deployment is complete.
5. Under **Post-deployment configuration script**, optionally specify the following parameters:
   - **Deployment settings** — Select the check box to ignore all deployment failures and proceed with a deployment.
   - **Configuration script** — Add one or more configuration scripts, depending on the number of servers included in the deployment. The scripts run in the order they are added. You can view detailed execution logs or failure information in the Amazon CloudWatch logs after a deployment is complete.
6. Choose **Next** when the preceding parameters are specified. You will be taken to the **Review infrastructure configuration** page.
7. Under **Infrastructure configuration**, specify the following parameters:
- **Key pair name** — Select a key pair to securely connect to your instance.
- **Virtual Private Cloud (VPC)** — The VPC in which the domain controllers will be deployed is the same as for the original deployment.
- **Private subnet** — The private subnet is determined by the subnet specified in your original deployment.
- **Security group assigned to additional application server (AAS)** — Select a security group that is currently assigned to a database node.

8. Choose **Next** when you are satisfied with your infrastructure configuration selections. You will be taken to the **Review** page.

9. On the **Review** page, verify your settings and configuration. Choose **Deploy** if you are satisfied with your selections. To edit your selections, choose **Edit** or **Previous**. It can take up to 10 minutes to create your new node.

### Manual activities required

When the automation workflow runs, the new node is attached to the existing SAP application and is reflected in the SAP console (SM51). The following manual activities are required to successfully add an AAS node to an existing Netweaver distributed installation. The following steps are presented as general guidelines and may not be complete for your scenario.

- Host entries are updated only on the PAS and newly added nodes. Refresh `/etc/hosts` entries from the PAS on all of the other existing nodes.
- Upload and set system profiles using transaction RZ10.
- Configure the number of work processes.
- Adjust or create logon and RFC server groups using transactions SMLG and RZ12.
- Adjust or create operation modes using transaction RZ04.

### Delete an addition application server (AAS) from an existing Netweaver distributed installation.

Before you delete an AAS node, verify that no users are logged in and no jobs are running on the instance.

**Note**

You can only delete a node that was created with the add or remove node feature using Launch Wizard for SAP.

### Pre- and post-deployment configuration scripts

As with base node deployments with Launch Wizard for SAP, you can run pre- and post- deployment configuration scripts when you add an additional node. For more information about how Launch Wizard for SAP accesses and deploys these scripts, see Custom deployment configuration scripts (p. 81).

### Limits for running pre- and post- deployment configuration scripts

The following limits apply when running pre- and post- deployment configuration scripts for your new node.

- Only one action can be performed at a time.
- SSM documents that are created at runtime are not deleted. Periodic cleanup of the documents is required.
- Any automation activity performed on the new node is extremely limited. Activities that must be performed on the parent node are called out in the SSM documentation.
Security groups in AWS Launch Wizard for SAP

This section describes the security groups that Launch Wizard for SAP creates and assigns to the database and application instances. It also describes how the entries in the outbound and inbound communication rules for database and application security groups are updated.

Topics
- Security groups (p. 148)
- Connectivity to external systems and users (p. 149)

Security groups

A security group acts as a virtual firewall that controls the traffic for one or more instances. When you allow Launch Wizard to create security groups, it creates a set of security groups and assigns them to the SAP database and application instances to allow for inbound traffic. Security groups use the following naming conventions:

- `<Infrastructure_Configuration_Name>_App_SecurityGroup`
- `<Infrastructure_Configuration_Name>_DB_SecurityGroup`

`<Infrastructure_Configuration_Name>_App_SecurityGroup`

`<Infrastructure_Configuration_Name>_App_SecurityGroup` is configured as follows to allow inbound access to the database servers.

<table>
<thead>
<tr>
<th>Source</th>
<th>Protocol</th>
<th>Port Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>All instances attached to this security group</td>
<td>all</td>
<td></td>
</tr>
<tr>
<td>All instances attached to the DB security group</td>
<td>TCP</td>
<td>1-65535</td>
</tr>
</tbody>
</table>

This configuration allows:
- inbound communication on all TCP ports from all of the SAP application servers deployed using the same configuration name
- inbound communication on all TCP ports from all of the database servers deployed using the same configuration name.

`<Infrastructure_Configuration_Name>_DB_SecurityGroup`

`<Infrastructure_Configuration_Name>_DB_SecurityGroup` is configured as follows to allow inbound access to the database servers.

<table>
<thead>
<tr>
<th>Source</th>
<th>Protocol</th>
<th>Port Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>All instances attached to this security group</td>
<td>all</td>
<td></td>
</tr>
<tr>
<td>All instances attached to the App security group</td>
<td>TCP</td>
<td>1-65535</td>
</tr>
</tbody>
</table>
Connectivity to external systems and users

CIDR/IP address and security group entries are entered in the infrastructure configuration. This allows access to SAP systems by front end users and upstream/downstream systems that are running in that CIDR block, or by end users (IP address) or systems assigned to those security groups. Port ranges are included in the rule definition that allow inbound access so that you can reuse the infrastructure configuration and deploy SAP systems with an instance number 00 to 99. Each entry in the outbound and inbound communication rules for a database security group, created either by the service or provided by the user, are updated as follows.

<table>
<thead>
<tr>
<th>Source</th>
<th>Protocol</th>
<th>Port Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>All instances attached to the App security group</td>
<td>UDP</td>
<td>111</td>
</tr>
<tr>
<td>All instances attached to the App security group</td>
<td>UDP</td>
<td>2049</td>
</tr>
<tr>
<td>All instances attached to the App security group</td>
<td>UDP</td>
<td>4000-4002</td>
</tr>
</tbody>
</table>

This configuration allows:

- inbound communication on all TCP ports from all of the SAP database servers deployed using the same configuration name.
- inbound communication on all TCP ports from all of the SAP application servers deployed using the same configuration name.
- inbound communication on UDP 111, 2049 and 4000 to 4002 from all the SAP application servers deployed using the same configuration name.

Each entry in the outbound and inbound communication rules for the application security group, created either by the service or by the user, are updated as follows.

<table>
<thead>
<tr>
<th>Source</th>
<th>Protocol</th>
<th>Port Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>TCP</td>
<td>22</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>1128 - 1129</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>4300 - 4399</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>8000 - 8099</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>8443</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>30013 - 39913</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>30015 - 39915</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>30017 - 39917</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>30041 - 39941</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>30044 - 39944</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>50013 - 59914</td>
</tr>
</tbody>
</table>

Each entry in the outbound and inbound communication rules for the application security group, created either by the service or by the user, are updated as follows.
<table>
<thead>
<tr>
<th>Source</th>
<th>Protocol</th>
<th>Port Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>TCP</td>
<td>22</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>3200 - 3399</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>8080</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>8443</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>3600-3699</td>
</tr>
<tr>
<td>Input</td>
<td>TCP</td>
<td>4237</td>
</tr>
</tbody>
</table>

**Note**
When the deployment is complete, you can update the security group information by adjusting the port range and source information.

**Note**
Launch Wizard considers a security group that it created as a shared resource. It does not delete the security group if you delete a deployment or if a deployment is rolled back.

# Troubleshoot AWS Launch Wizard for SAP

Each application in your account in the same AWS Region can be uniquely identified by the application name specified at the time of a deployment. The application name can be used to view the details related to the application launch.

**Contents**
- Launch Wizard provisioning events (p. 150)
- CloudWatch Logs (p. 150)
- AWS CloudFormation stack (p. 151)
- Pre- and post-deployment configuration scripts (p. 151)
- Application launch quotas (p. 151)
- Enable termination protection (p. 151)
- Instance level logs (p. 151)
- SAP application software deployment logs (p. 152)
- Errors (p. 152)

**Launch Wizard provisioning events**

Launch Wizard captures events from SSM Automation and AWS CloudFormation to track the status of an ongoing application deployment. If an application deployment fails, you can view the deployment events for this application by selecting **Deployments** from the navigation pane. A failed event shows a status of **Failed** along with a failure message.

**CloudWatch Logs**

Launch Wizard streams provisioning logs from all of the AWS log sources, such as AWS CloudFormation, SSM, and CloudWatch Logs. CloudWatch Logs for a given application name can be viewed on the CloudWatch console for the log group name **LaunchWizard-APPLICATION_NAME** and log stream **ApplicationLaunchLog**.
AWS CloudFormation stack

Launch Wizard uses AWS CloudFormation to provision the infrastructure resources of an application. AWS CloudFormation stacks can be found in your account using the AWS CloudFormation describe-stacks API. Launch Wizard launches various stacks in your account for validation and application resource creation. The following are the relevant filters for the describe-stacks API.

- **Application resources**
  
  LaunchWizard-APPLICATION_NAME.

You can view the status of these AWS CloudFormation stacks. If any of them fail, you can view the cause of the failure.

Pre- and post-deployment configuration scripts

Can't find the output of my scripts

- **Cause:** Customizations are key scripts that you want to run on the EC2 instances and the logs from script deployments are not included with the provisioning logs.
- **Solution:** The logs for scripts that run on EC2 instances are included in the CloudWatch log group that Launch Wizard creates in your account for the workload. The CloudWatch log group can be identified as LaunchWizard-APPLICATION_NAME. You can find the following logs in this log group.
  - lw-customization/<instance-id>/preDeploymentConfiguration — For pre-deployment configuration scripts that run on the specified EC2 instance.
  - lw-customization/<instance-id>/postDeploymentConfiguration — For post-deployment configuration scripts that run on the specified EC2 instance.

Application launch quotas

Launch Wizard allows for a maximum of 25 active applications for any given application type. Up to three applications can be in progress at a time. If you want to increase this limit, contact AWS Support.

Enable termination protection

If you encounter errors when you deploy SAP systems with Launch Wizard, and the log information provided by Amazon CloudWatch is not sufficient to determine your issue, you must log in to the instance to determine the root cause. In case of failures, Launch Wizard terminates the instances, which limits the means to troubleshoot a failure.

You can update the termination settings to disable termination of the instances from the EC2 console. From the Instances page, select an instance and choose **Action > Instance Settings > Change Termination Protection**. Then choose **Yes, Enable**.

After you have determined the root cause, disable the termination protection before you delete the deployment in Launch Wizard.

Instance level logs

Detailed deployment and configuration logs can be found in /root/install/scripts/log during the deployment process, when the instances are being launched and configured. The name of the log...
file is `install.log`. To check the progress of the deployment, you can log in to an instance as soon its instance state is listed as **running**.

When the deployment is finished, the log files are moved to `/tmp`.

In addition to the `install.log` file, the `install.dbg` file includes additional error information in case of failure scenarios. `install.json` includes all of your console selections to configure the deployment.

### SAP application software deployment logs

Depending on which SAP components are deployed on an instance, Launch Wizard creates a folder in `/tmp` to log all of the SAP software application deployment logs. If a database component is deployed on an instance, the folder name in the file will be `NW_ABAP_DB`. If an application server is deployed, the folder name will be `NW_ABAP_APP`. For single node deployments, there will be multiple folders, such as `NW_ABAP_DB` and `NW_ABAP_CI`, which represent the different components deployed on the instance.

### Errors

**Your requested instance type is not supported in your requested Availability Zone**

- **Cause**: This failure might occur during the launch of your instance, or during the validation of the instances that Launch Wizard launches in your selected subnets.
- **Solution**: For this scenario, you must choose a different Availability Zone and retry the deployment from the initial page of the Launch Wizard console.

**Infrastructure template already exists**

- **Cause**: This failure occurs when you choose to create a new infrastructure configuration and then navigate back to the first step in the wizard to review or adjust any settings. Launch Wizard has already registered the configuration template, so choosing **Next** results in the error "Template name already exists. Select a new template name."
- **Solution**:
  
  Perform one of the following actions to continue with your deployment.
  
  - Change the name of the configuration template and continue.
  - Choose another template and continue.
  - Delete the template causing the error by navigating to the **Saved Infrastructure Setting** tab under **Deployments – SAP**, and then continue with your configuration using the same configuration name.
AWS Launch Wizard for SQL Server

What Is AWS Launch Wizard for SQL Server?

AWS Launch Wizard is a service that guides you through the sizing, configuration, and deployment of Microsoft SQL Server applications on AWS, following the AWS Well-Architected Framework. AWS Launch Wizard supports both single instance and high availability (HA) application deployments.

AWS Launch Wizard reduces the time it takes to deploy SQL Server solutions to the cloud. You input your application requirements, including performance, number of nodes, and connectivity, on the service console. AWS Launch Wizard identifies the right AWS resources to deploy and run your SQL Server application. AWS Launch Wizard provides an estimated cost of deployment, and you can modify your resources and instantly view the updated cost assessment. When you approve, AWS Launch Wizard provisions and configures the selected resources in a few hours to create a fully-functioning production-ready SQL Server application. It also creates custom AWS CloudFormation templates, which can be reused and customized for subsequent deployments.

Once deployed, your SQL Server application is ready to use and can be accessed on the EC2 console. You can manage your SQL Server application with AWS SSM.

Contents

- Supported operating systems and SQL versions (p. 154)
- Features of AWS Launch Wizard (p. 154)
- Components (deployment on Windows) (p. 157)
- Components (deployment on Linux) (p. 159)
- Related services (p. 160)
- How AWS Launch Wizard works (p. 161)
- Deployment options (p. 165)
Supported operating systems and SQL versions

AWS Launch Wizard supports the following operating systems and SQL Server versions:

Deployments on Windows

- Windows Server 2019/2016/2012 R2

Amazon FSx for Failover Clustering (FCI) deployments on Windows

- Enterprise and Standard Editions of Microsoft SQL Server 2019/2017/2016 SP2

CUs are installed at the same time as public AMIs for SQL license-included AMIs. CUs and service packs are not installed for license-included Windows AMIs and BYOL AMIs.

Deployments on Ubuntu

- Ubuntu 18.04
- Enterprise and Standard Edition of Microsoft SQL Server 2019

Deployments on RHEL

- Red Hat Enterprise Linux (RHEL) 7.9
- Enterprise and Standard Edition of Microsoft SQL Server 2019/2017

Features of AWS Launch Wizard

AWS Launch Wizard provides the following features:

- Simple application deployment (p. 154)
- AWS resource selection (p. 155)
- Cost estimation (p. 155)
- Reusable code templates (p. 155)
- SNS notification (p. 155)
- Always On Availability Groups (SQL Server) (p. 155)
- Dedicated Hosts (deployment on Windows) (p. 155)
- Early input validation (p. 155)
- Application resource groups for easy discoverability (p. 156)
- One-click monitoring (p. 156)
- Amazon FSx for Failover Clustering (FCI) (p. 157)

Simple application deployment

AWS Launch Wizard makes it easy for you to deploy third-party applications on AWS, such as Microsoft SQL Server. When you input the application requirements, AWS Launch Wizard deploys the necessary
AWS resources for a production-ready application. This means that you do not have to manage separate infrastructure pieces or spend time provisioning and configuring your SQL Server application.

**AWS resource selection**

Launch Wizard considers performance, memory, bandwidth, and other application features to determine the best instance type, EBS volumes, and other resources for your SQL Server application. You can modify the recommended defaults.

**Cost estimation**

Launch Wizard provides a cost estimate for a complete deployment. The cost estimate is itemized for each individual resource to deploy. The estimated cost automatically updates each time you change a resource type configuration in the wizard. The provided estimates are for general comparisons only. The estimates are based on On-Demand costs and actual costs may be lower.

**Reusable code templates**

Launch Wizard creates a CloudFormation stack that can be reused to customize and replicate your infrastructure in multiple environments. Code in the template helps you provision resources. You can access and use the templates created by your Launch Wizard deployment from the CloudFormation console. For more information about CloudFormation stacks, see Working with stacks.

**SNS notification**

You can provide an SNS topic so that Launch Wizard will send you notifications and alerts about the status of a deployment.

**Always On Availability Groups (SQL Server)**

Always On Availability Groups (AG) is a Microsoft SQL Server feature that is supported by the AWS SQL Server installation. AG augments the availability of a set of user databases. An availability group supports a failover environment for a discrete set of user databases, known as availability databases. If one of these databases fails, another database takes over its workload with no impact on availability. Always On Availability improves database availability, enabling more efficient resource usage. For more information about the concepts and benefits of Always On Availability, see Always On Availability Groups (SQL Server).

**Dedicated Hosts (deployment on Windows)**

You can deploy SQL Server Always On Availability Groups (AG) or basic availability groups on your Dedicated Hosts to leverage your existing SQL Server Licenses (BYOL). From the Launch Wizard console, select **Dedicated Host** tenancy, and then select the Dedicated Hosts for your VPC. For more information about Amazon EC2 Dedicated Hosts, see Dedicated Hosts.

**Early input validation**

You can leverage your existing infrastructure (such as VPC or Active Directory) with Launch Wizard. This may lead to deployment failures if your existing infrastructure does not meet certain deployment prerequisites. For example, for a SQL Server Always On deployment in your existing VPC, the VPC must have at least one public subnet and two private subnets. It must also have outbound connectivity to Amazon S3, Systems Manager, and AWS CloudFormation service endpoints. If these requirements are not met, the deployment will fail. If you are in a later stage of a deployment, this failure can take more than an hour to detect. To detect these types of issues early in the application deployment process, Launch Wizard's validation framework verifies key application and infrastructure specifications before
provisioning. Verification takes approximately 15 minutes. If necessary, you can take appropriate actions to adjust your VPC configuration.

Launch Wizard performs the following infrastructure validations:

**Resource limit validations at the AWS account level:**

- VPC
- Internet gateway
- Number of CloudFormation stacks

**Additionally, Launch Wizard performs the following application-specific validations:**

- Active Directory credentials (deployment on Windows)
- Public subnet outbound connectivity
- Private subnet outbound connectivity
- Custom Windows AMIs:
  - SQL Server installed and running on instance
  - Compliant versions of Windows and SQL Server
- Dedicated Hosts (deployment on Windows)
  - AMIs are filtered according to the billing code. When you select Dedicated Host tenancy in the application, the AMI selection dropdown list filters out AMIs for which the usage operation is set to include SQL Server Enterprise or SQL Server Standard, per the [details and usage operation values](#). This filtering behavior is the result of restrictions described in the [Dedicated Host restrictions](#) page.
  - Supported instance type
  - Sufficient capacity to launch number of nodes and instances
  - Selected subnet and corresponding Dedicated Host are in the same Availability Zone for any additional nodes beyond the primary and first secondary nodes

**Note**

Some validations, for example for valid Active Directory credentials, require Application Wizard to launch a t2.large EC2 instance in your account for a few minutes. After it runs the necessary validations, Launch Wizard terminates the instance.

**Application resource groups for easy discoverability**

Launch Wizard creates a resource group for all of the AWS resources created for your SQL Server application. You can manage the resources through the EC2 console or with Systems Manager. When you access Systems Manager through Launch Wizard, the resources are automatically filtered for you based on your resource group. You can manage, patch, and maintain your SQL Server applications in Systems Manager.

**One-click monitoring**

Launch Wizard integrates with [CloudWatch Application Insights](#) to provide a one-click monitoring setup experience for deploying SQL Server HA workloads on AWS. When you select the option to set up monitoring and insights with Application Insights on the Launch Wizard console, Application Insights automatically sets up relevant metrics, logs, and alarms on CloudWatch, and starts monitoring newly deployed workloads. You can view automated insights and detected problems, along with the health of your SQL Server HA workloads, on the CloudWatch console.

Counters that you can configure using Application Insights include:
• Mirrored Write Transaction/sec
• Recovery Queue Length
• Transaction delay
• Windows Event Logs on CloudWatch

You can also get automated insights when a failover event or problem, such as a restricted access to query a target database, is detected on your workload.

Amazon FSx for Failover Clustering (FCI)

Launch Wizard uses Amazon FSx to provide Failover Clustering for SQL Server deployments. Failover Clustering is a high availability solution for SQL that puts all database and log files in shared storage (Amazon FSx). The Amazon FSx file share spans multiple Availability Zones and is highly redundant, allowing for automatic failover between SQL nodes in the event of failure.

Components (deployment on Windows)

A SQL Server application deployed on Windows with Launch Wizard includes the following components:

• A virtual private cloud (VPC) configured with public and private subnets across two Availability Zones. A public subnet is a subnet whose traffic is routed to an internet gateway. If a subnet does not have a route to the internet gateway, then it is a private subnet. The VPC provides the network infrastructure for your SQL Server deployment. You can choose an optional third Availability Zone for additional SQL cluster nodes, as shown below.
• An internet gateway to provide access to the internet.
• In the public subnets, Windows Server-based Remote Desktop Gateway (RDGW) instances and network address translation (NAT) gateways for outbound internet access. If you are deploying in your preexisting VPC, Launch Wizard uses the existing NAT gateway in your VPC. For more information about NAT gateways, see NAT Gateways.
• Elastic IP addresses associated with the NAT gateway and RDGW instances. For more information about Elastic IP addresses, see Elastic IP Addresses.
• In the private subnets, Active Directory domain controllers.
• In the private subnets, Windows Server-based instances as Windows Server Failover Clustering (WSFC) nodes. For more information, see Windows Server Failover Clustering with SQL Server.
• SQL Server Enterprise edition with SQL Server Always On Availability Groups on each WSFC node. This architecture provides redundant databases and a witness server to ensure that a quorum can vote for the node to be promoted to the controlling resource. The default architecture mirrors an on-premises architecture of two SQL Server instances spanning two subnets placed in two different Availability Zones. For more information about SQL Server Always On Availability Groups, see Overview of Always On Availability Groups (SQL Server).
• Security groups to ensure the secure flow of traffic between the instances deployed in the VPC. For more information, see Security Groups for Your VPC.

Note
If you choose to deploy SQL Server Always On through Launch Wizard into your existing VPC, there is an additional mandatory check box on the console to indicate whether VPC and public/private subnet requirements have been met.

• Amazon FSx to provide highly available and redundant storage across Availability Zones for clustering.

Note
Launch Wizard uses two Availability Zones.

You can build a SQL HA installation, as shown in the following diagram.
You can also choose to build an architecture with SQL Server Always On FCI, as shown in the following diagram.
Components (deployment on Linux)

A SQL Server application deployed on Linux with Launch Wizard includes the following components:

- **A virtual private cloud (VPC)** configured with public and private subnets across three Availability Zones. A public subnet is a subnet whose traffic is routed to an internet gateway. If a subnet does not have a route to the internet gateway, then it is a private subnet. The VPC provides the network infrastructure for your SQL Server deployment.

- **An internet gateway** to provide access to the internet.

- In the public subnets, **network address translation (NAT)** for outbound internet access. If you are deploying in your preexisting VPC, Launch Wizard uses the existing NAT gateway in your VPC. For more information about NAT gateways, see NAT Gateways.

- Two of the private subnets each run a SQL Server **replica node**. One acts as the primary node, and the other as secondary node. The third private subnet is used to run the configuration replica. Launch Wizard deployments on Linux use Pacemaker as the cluster resource manager. Pacemaker differs from Windows Server Failover Cluster (WSFC), which is used for Windows deployments, in terms of how it handles quorum. For Always On availability groups (AG) on Linux, arbitration happens in SQL Server where the metadata is stored. This is where the configuration-only replica is relevant. In order to maintain quorum and enable automatic failovers, Launch Wizard sets up a third node that acts as the configuration-only replica.

- **Security groups** to ensure the secure flow of traffic between the instances deployed in the VPC. For more information, see Security Groups for Your VPC.

The high-level architecture of a SQL Server high availability solution on Linux is similar to the architecture for deployment on Windows. The main differences are the low-level components and technologies. The architecture for Linux deployments provides redundant databases and a configuration-only replica node to verify that a quorum can vote for the node to be promoted to the controlling resource. The default architecture mirrors an on-premises architecture of two SQL Server instances spanning two subnets placed in two different Availability Zones. For more information about SQL Server Always On Availability Groups (AG), see Overview of Always On Availability Groups (SQL Server) in the Microsoft documentation.
The following services are used when you deploy a SQL Server application with AWS Launch Wizard:

- AWS CloudFormation (p. 160)
- AWS SSM (p. 161)
- Amazon Simple Notification Service (SNS) (p. 77)
- Amazon CloudWatch Application Insights (p. 161)
- Linux-only technologies (p. 161)

**AWS CloudFormation**

AWS CloudFormation is a service for modeling and setting up your AWS resources, enabling you to spend more time focusing on your applications that run in AWS. You create a template that describes all of the AWS resources that you want to use (for example, Amazon EC2 instances or Amazon RDS DB instances), and AWS CloudFormation takes care of provisioning and configuring those resources for you. With Launch Wizard, you don’t have to sift through CloudFormation templates to deploy your application. Instead, Launch Wizard combines infrastructure provisioning and configuration (with a CloudFormation template) and application configuration (with code that runs on EC2 instances to configure the application) into a unified SSM Automation document. The SSM document is then invoked by Launch Wizard’s backend service to provision a SQL Server application in your account. For more information, see the AWS CloudFormation User Guide.
AWS SSM

**AWS SSM** is a collection of capabilities for configuring and managing your Amazon EC2 instances, on-premises servers and virtual machines, and other AWS resources at scale. Systems Manager includes a unified interface that enables you to centralize operational data and automate tasks across your AWS resources. Systems Manager shortens the time to detect and resolve operational problems in your infrastructure. You have the option of managing your application with Systems Manager after deploying with Launch Wizard. For more information, see the *AWS Systems Manager User Guide*.

Amazon Simple Notification Service (SNS)

**Amazon Simple Notification Service (SNS)** is a highly available, durable, secure, fully managed pub/sub messaging service that provides topics for high-throughput, push-based, many-to-many messaging. Using Amazon SNS topics, your publisher systems can fan out messages to a large number of subscriber endpoints and send notifications to end users using mobile push, SMS, and email. You can use SNS topics for your Launch Wizard deployments to stay up-to-date on deployment progress. For more information, see the *Amazon Simple Notification Service Developer Guide*.

Amazon CloudWatch Application Insights

**Amazon CloudWatch Application Insights** facilitates observability for .NET and SQL Server applications. It can help you set up the best monitors for your application resources to continuously analyze data for signs of problems with your applications. Application Insights, which is powered by Sagemaker and other AWS technologies, provides automated dashboards that show potential problems with monitored applications, helping you to quickly isolate ongoing issues with your applications and infrastructure. The enhanced visibility into the health of your applications that Application Insights provides can help you reduce your mean time to repair (MTTR) so that you don't have to pull in multiple teams and experts to troubleshoot your application issues.

Linux-only technologies

The following key technologies are used when you deploy a SQL Server application with Amazon Launch Wizard to the Linux platform.

- **Pacemaker** is an open source cluster resource manager (CRM), which is a system that coordinates managed resources and services made highly available by a cluster.
- **Corosync** is an open source program that provides cluster membership and messaging capabilities, often referred to as the messaging layer, to client servers. In contrast to Pacemaker, which allows you to control cluster behavior, Corosync makes it possible for servers to communicate as a cluster.
- **Transact-SQL** is an extension to the SQL language. It is used to interact with relational databases. Transact-SQL is platform-agnostic and can be used to configure the AlwaysOn Availability Group and listener.
- **Fencing** is used to isolate a malfunctioning server from the cluster in order to protect and secure the synced resources. The recommended solution to use in the case of a malfunctioning server is the "Shoot the other node in the head" (STONITH) method. STONITH is a fencing technique that isolates a failed node so that it does not disrupt a computer cluster. The STONITH method fences failed nodes by resetting or powering down the failed node. Fencing is also used when a clustered service cannot be stopped. In this case, the cluster uses fencing to force the whole node offline, which makes it safe to start the service from a different server. Fencing can be performed at two levels: the node or resource level. Launch Wizard only supports node-level fencing.

How AWS Launch Wizard works

AWS Launch Wizard provides a complete solution to provision popular third-party applications on AWS. Currently, Launch Wizard supports Microsoft SQL Server application deployments across multiple
Availability Zones or on a single instance. You provide the specifications, such as for performance, throughput, and networking. Based on the application requirements that you enter, Launch Wizard automatically provisions the right AWS resources in the cloud. For example, Launch Wizard determines the best instance type and EBS volume for your CPU, memory, and bandwidth specifications, then deploys and configures them.

Launch Wizard provides an estimated cost of deployment. You can modify your resources and instantly view an updated cost assessment. Once you approve, Launch Wizard validates the inputs and flags inconsistencies. When the inconsistencies are resolved, Launch Wizard provisions the resources and configures them. The result is a ready-to-use SQL Server Always On application.

Launch Wizard creates a CloudFormation stack according to your infrastructure needs. You can reuse the template created by CloudFormation as a baseline for future infrastructure provisioning.

For deployments on Windows, Launch Wizard supports AWS Managed Microsoft Active Directory (AD). It also supports connecting to on-premises Active Directory using AWS Direct Connect.

**Topics**
- Implementation details for deployment on Windows (p. 162)
- Implementation details for deployment on Linux (p. 164)

**Implementation details for deployment on Windows**

AWS Launch Wizard implements SQL Server deployments on Windows as follows:
- SQL Server Enterprise Edition (p. 153)
- Storage on WSFC nodes (p. 162)
- IP Addressing on the Windows Server Failover Clustering (WSFC) nodes (p. 163)
- Windows Server Failover Clustering (WSFC) (p. 163)
- Always On configuration (p. 163)
- Failover clustering (p. 164)

**SQL Server Enterprise Edition**

Launch Wizard supports installation of SQL Server Enterprise and Standard Editions of 2016 and 2017 on Windows Server 2012 R2, 2016, and 2019 through license-included Amazon Machine Images (AMIs). Launch Wizard allows you to bring your own SQL licenses through a custom AMI. If you use a custom AMI, ensure that your AMI meets the requirements listed in Requirements for using custom AMIs (deployment on Windows) (p. 170).

**Storage on WSFC nodes**

Storage capacity and performance are key aspects of any production SQL Server installation. Launch Wizard lets you choose capacity and performance based on your deployment requirements.

Amazon Elastic Block Store (Amazon EBS) volumes are included in the architecture to provide durable, high-performance storage. EBS volumes are network-attached disk storage, which you can create and attach to EC2 instances. When attached to an instance, you can create a file system on top of the volumes, run a database, or use them in any way that you would use a block device. EBS volumes are placed in a specific Availability Zone, where they are automatically replicated to protect you from the failure of a single component. EBS volume type io1 is not supported.

**Provisioned IOPS EBS volumes** offer storage with consistent and low-latency performance. They are backed by solid state drives (SSDs) and designed for applications with I/O intensive workloads, such as databases. Amazon EBS-optimized instances, such as the R4 instance type, deliver dedicated throughput between Amazon EC2 and Amazon EBS.
By default, Launch Wizard deploys three 500 GiB general purpose SSD volumes to store databases, logs, tempdb files, and backups on each WSFC node. These general purpose SSD volumes are in addition to the root general purpose SSD volume used by the operating system, which delivers a consistent baseline of 3 IOPS/GiB and provides a total of 1,500 IOPS per volume for SQL Server database and log volumes. You can customize the volume size and switch to using dedicated IOPS volumes with the volume you specify. If you need more IOPS per volume, consider using Provisioned IOPS SSD volumes by changing the SQL Server Volume Type and SQL Server Volume IOPS parameters.

The default disk layout for SQL Server deployed by Launch Wizard is:

- One general purpose SSD volume (100 GiB) for the operating system (C:)
- One general purpose SSD volume (500 GiB) to host the SQL Server database files (D:)
- One general purpose SSD volume (500 GiB) to host the SQL Server log files (E:)
- One general purpose SSD volume (500 GiB) to host the SQL Server tempdb and backup files (F:)

**IP Addressing on the Windows Server Failover Clustering (WSFC) nodes**

In order to support WSFC and Always On Availability Group listeners, each node that hosts the SQL Server instances that participate in the cluster must have three IP addresses assigned, as follows:

- One IP address as the primary IP address for the instance
- A second IP address as the WSFC IP resource
- A third IP address to host the Always On Availability Group listener

When you launch the AWS CloudFormation template, you can specify the addresses for each node. By default, the underlying CloudFormation templates of Launch Wizard use 10.0.0.0/20, 10.0.16.0/20, and 10.0.32.0/20 as CIDR blocks for the private subnets. This is true only when you use Launch Wizard to deploy SQL Server Always On clusters in a new VPC.

**Windows Server Failover Clustering (WSFC)**

You can build the failover cluster after your Windows Server instances have been deployed and domain-joined. Launch Wizard's underlying AWS CloudFormation templates build the cluster when deploying the second node. If you use the default template parameter settings, Launch Wizard executes the following Windows PowerShell commands to complete this task.

```
PS C:\> Install-WindowsFeature failover-clustering -IncludeManagementTools
PS C:\> New-Cluster -Name WSFClusterName -Node $nodes -StaticAddress $addr
```

The first command runs on each instance during the bootstrapping process. It installs the required components and management tools for the failover clustering services. The second command runs near the end of the bootstrapping process on the second node and is responsible for creating the cluster and for defining the server nodes and IP addresses.

If you set the optional third Availability Zone, Launch Wizard keeps the quorum settings to the default node majority.

```
PS C:\> Set-ClusterQuorum -NodeMajority
```

**Always On configuration**

After SQL Server Enterprise edition has been installed and the Windows Server failover cluster has been built, Launch Wizard enables SQL Server Always On with the following PowerShell command.
Launch Wizard runs this command on each node, and the proper server name is provided as a value for the `ServerInstance` parameter.

When the deployment is complete, Launch Wizard creates your databases and make them highly available by creating an Always On Availability Group.

When you create an availability group, you provide a network share that is used to perform an initial data synchronization. As you progress through the New Availability Group Wizard, a full backup for each selected database is created and placed in the share. The secondary node connects to the share and restores the database backups before joining the availability group.

### Failover clustering

Launch Wizard sets up and installs SQL Server failover clustering on Windows on each of the nodes that uses PowerShell DSC modules, which is the first step it takes to configure the cluster. Next, Launch Wizard adds the specified account as the domain account to the cluster group.

You can use either license-included AMIs or your own custom AMIs. If you use license-included AMIs, they must be Windows or SQL license-included AMIs.

### Implementation details for deployment on Linux

AWS Launch Wizard implements SQL Server deployments on Linux as follows.

#### Always On availability group configuration

- **Availability group enablement.** Launch Wizard enables availability groups using the following commands.

  ```
  sudo /opt/mssql/bin/mssql-conf set hadr.hadrenabled 1
  sudo systemctl restart mssql-server
  ```

- **Authentication of TCP endpoints.** An availability group uses TCP endpoints for communication. Launch Wizard uses certificate-based authentication to support endpoints for availability groups. All nodes create a self-signed certificate and upload their certificates to an Amazon S3 location that you specify. Each node will then locally sync the certificates from all of the other nodes.

- **Creation of availability group endpoints.** Launch Wizard performs the backup, create, and restore of the authentication certificates with Transact-SQL on Linux. Transact-SQL on Linux is also used to create the availability group endpoints and availability groups with automatic seeding. The listener and read-only routing are configured after an availability group is created.

- **Creation of availability group resources in Pacemaker cluster.** After an availability group is created in SQL Server, the corresponding resources must be created in Pacemaker. There are two resources associated with an availability group: the availability group and a floating IP address. The created availability group resource is a unique resource called a clone. The availability group resource contains copies on each node, with one controlling resource. The controlling resource is associated with the server that hosts the primary replicas. The secondary replicas (except for configuration-only replicas) can be promoted to controlling resource during failover.

  The floating IP address is an unused IP address in the VPC that is not part of any of the subnet CIDR ranges that are part of this cluster. Launch Wizard creates a DNS entry in the host files to map AGLListenerName to a floating IP Address. Launch Wizard also creates a route for Floating IP/32 to the
Primary Node Network Interface so that all internal VPC network traffic is routed to the primary node when any traffic reaches the floating IP. In case of failover, the resource agent will update the routing table to point the floating IP to the new primary node.

• **Creation of availability group listener.** The creation of the availability group listener is performed using the following Transact-SQL commands:

```
ALTER AVAILABILITY GROUP MyAg2
ADD LISTENER ...
```

### Deployment options

**AWS Launch Wizard provides the following deployment paths:**

- Deployment on Windows (p. 165)
- Deployment on Linux (p. 166)

#### Deployment on Windows

1. **Deploy SQL Server into a new VPC across multiple Availability Zones.**

   When you choose this configuration option, Launch Wizard builds a new AWS environment consisting of the VPC, subnets, NAT gateways, security groups, domain controllers, and other infrastructure components. It then deploys Windows Server Failover Clustering (WSFC) with SQL Server across multiple Availability Zones into the new VPC.

2. **Deploy SQL Server into a new VPC on a single node.**

   When you choose this configuration option, Launch Wizard builds a new AWS environment consisting of the VPC, subnet, NAT gateway, security groups, domain controllers, and other infrastructure components. It then deploys SQL Server on a single node into the new VPC.

3. **Deploy SQL Server into an existing VPC and create a new AWS Managed Active Directory.**

   When you choose this configuration option, Launch Wizard builds a new AWS environment that consists of security groups, domain controllers, and other infrastructure components, and then deploys Windows Server Failover Clustering (WSFC) with SQL Server into the specified VPC and subnets. Your AWS environment must include a VPC with two or three Availability Zones, private subnets in each Availability Zone, and at least one public subnet in the VPC. Launch Wizard supports only **AWS Managed Microsoft Active Directory** for this scenario.

4. **Deploy SQL Server into an existing VPC with an existing AWS Managed Active Directory across multiple Availability Zones.**

   When you choose this configuration option, Launch Wizard provisions Windows Server Failover Clustering (WSFC) across multiple Availability Zones in your existing AWS infrastructure. Your AWS environment must include a VPC with two or three Availability Zones, private subnets in each Availability Zone, at least one public subnet in the VPC, and an AWS Active Directory in the VPC (this is the Active Directory on which you deploy your SQL nodes).

5. **Deploy SQL Server into an existing VPC with an existing AWS Managed Active Directory on a single node.**

   When you choose this configuration option, Launch Wizard provisions SQL Server on a single node in your existing AWS infrastructure. Your AWS environment must include a VPC in one Availability Zone, a private subnet, a public subnet in the VPC, and an AWS Active Directory in the VPC (this is the Active Directory on which you deploy your SQL nodes).
6. **Deploy SQL Server into an existing VPC across multiple Availability Zones and connect to an on-premises Active Directory.**

   When you choose this configuration option, Launch Wizard provisions Windows Server Failover Clustering (WSFC) across multiple Availability Zones in your existing AWS infrastructure. Your AWS environment must include a VPC with two or three Availability Zones, private subnets in each Availability Zone, at least one public subnet in the VPC, and an AWS Direct Connect connection to your on-premises Active Directory.

7. **Deploy a SQL Server into an existing VPC on a single node and connect to an on-premises Active Directory.**

   When you choose this configuration option, Launch Wizard provisions SQL Server on a single node in your existing AWS infrastructure. Your AWS environment must include a VPC in one Availability Zone, a private subnet, a public subnet in the VPC, and an AWS Direct Connect connection to your on-premises Active Directory.

8. **Deploy SQL HA on Dedicated Hosts with your Windows BYOL or SQL Server BYOL licensed AMIs.**

   When you choose this configuration option, Launch Wizard provisions SQL Server Always On Availability Groups on your existing Dedicated Hosts using your existing SQL Server licenses (BYOL) for deploying SQL Server High Availability solutions.

You can configure additional settings, such as the SQL Server version (by your choice of AMI), in addition to instance types and Amazon EBS volume types based on the infrastructure requirements that you specify.

### Deployment on Linux

1. **Deploy SQL Server into a new VPC across multiple Availability Zones.**

   When you choose this configuration option, Launch Wizard builds a new AWS environment consisting of the VPC, subnets, NAT gateways, security groups, and other infrastructure components. It then deploys SQL Server Always On Availability Groups (AG) across multiple Availability Zones into the VPC using Pacemaker and fencing agents for cluster management.

2. **Deploy SQL Server into a new VPC on a single node.**

   When you choose this configuration option, Launch Wizard builds a new AWS environment consisting of the VPC, subnet, NAT gateways, security groups, and other infrastructure components. It then deploys SQL Server AG on a single node into the VPC using Pacemaker and fencing agents for cluster management.

3. **Deploy SQL Server into an existing VPC across multiple Availability Zones.**

   When you choose this configuration option, Launch Wizard builds a new AWS environment that consists of security groups and other infrastructure components, and then deploys SQL Server AG across multiple Availability Zones into the specified VPC and subnets. Your AWS environment must include a VPC with three Availability Zones, private subnets in each Availability Zone, and at least one public subnet in the VPC.

4. **Deploy SQL Server into an existing VPC on a single node.**

   When you choose this configuration option, Launch Wizard builds a new AWS environment that consists of security groups and other infrastructure components, and then deploys SQL Server AG on a single node into the specified VPC and subnet. Your AWS environment must include a VPC with one Availability Zone, a private subnet, and a public subnet in the VPC.
Default quotas

Launch Wizard allows for a maximum of 50 active applications (with status in progress or completed) for any given application type. If you want to increase this limit, contact AWS Support. Launch Wizard supports three parallel, in-progress deployments per account.

Get started with AWS Launch Wizard for SQL Server

This section contains information to help you set up your environment to deploy SQL Server with Launch Wizard, including:

- Active Directory permissions
- How to create an IAM policy and attach it to your IAM user identity
- OS and SQL version requirements
- Configuration settings

When your environment is set up, you can deploy a SQL Server Always On application with Launch Wizard by following the steps and parameter specification details (p. 173) provided in this section.

Topics to help you get started:
- Set up for AWS Launch Wizard for SQL Server (p. 167)
- Deploy an application with AWS Launch Wizard for SQL Server on Windows (p. 173)
- Deploy an application with AWS Launch Wizard for SQL Server on Ubuntu (p. 185)
- Deploy an application with AWS Launch Wizard for SQL Server on RHEL (p. 192)

Set up for AWS Launch Wizard for SQL Server

The following prerequisites must be verified to deploy a SQL Server Always On application with AWS Launch Wizard.

- Active Directory (Windows deployment) (p. 167)
- AWS Identity and Access Management (IAM) (p. 169)
- AWS Secrets Manager (p. 169)
- Requirements for using custom AMIs (deployment on Windows) (p. 170)
- Requirements for using custom AMIs (deployment on Linux) (p. 171)
- Requirements for using Windows license-included AMIs (SQL Server FCI configuration only) (p. 171)
- Requirements for using Amazon FSx (p. 171)
- Configuration settings (deployment on Windows) (p. 172)

Active Directory (Windows deployment)

AWS Managed Active Directory

If you are deploying SQL Server into an existing VPC with an existing Active Directory, Launch Wizard uses your Managed Active Directory (AD) domain user credentials to set up a fully functional SQL Server Always On Availability Group in the Active Directory. Launch Wizard supports this deployment option.
only for AWS Managed Active Directory. Your Managed Active Directory does not have to be in the same VPC as the one in which SQL Server Always On is deployed. If it is in a different VPC than the one in which SQL Server Always On is deployed, verify that you set up connectivity between the two VPCs. The domain user requires the following permissions in the Active Directory Default organizational unit (OU) to enable Launch Wizard to perform the deployment successfully:

- Reset password
- Write userAccountControl
- Create user accounts
- Create computer objects
- Read all properties
- Modify permissions

The following key operations are performed against your Active Directory by Launch Wizard. These operations result in the creation of new records or entries in Active Directory.

- SQL Server service user added as a new Active Directory user if it does not already exist in Active Directory.
- SQL Server instance and Remote Desktop Gateway Access instance joined to the Active Directory domain.
- CreateChild role added to Windows Server Failover Cluster as part of ActiveDirectoryAccessRule.
- FullControl role added to SQL Server Service user as part of FileSystemRights.

On-premises Active Directory through AWS Direct Connect

If you are deploying SQL Server into an existing VPC and connecting to an on-premises Active Directory, verify the following prerequisites.

- Make sure you have connectivity between your AWS account and your on-premises network. You can establish a dedicated network connection from your on-premises network to your AWS account with AWS Direct Connect. For more information, see the AWS Direct Connect documentation.
- The domain functional level of your Active Directory domain controller must be Windows Server 2012 or later.
- The IP addresses of your DNS server must be either in the same VPC CIDR range as the one in which your Launch Wizard SQL Server Always On deployment will be created, or in the private IP address range.
- The firewall on the Active Directory domain controllers should allow the connections from the Amazon VPC from which you will create the Launch Wizard deployment. At a minimum, your configuration should include the ports mentioned in How to configure a firewall for Active Directory domains and trusts.
- The domain user requires the following permissions in the Active Directory Default organizational unit (OU) to enable Launch Wizard to perform the deployment successfully:

  - Reset password
  - Write userAccountControl
  - Create user accounts
  - Create computer objects
  - Read all properties
  - Modify permissions

You can optionally perform the following step.
• Establish DNS resolution across your environments. For options on how to set this up, see How to Set Up DNS Resolution Between On-Premises Networks and AWS Using AWS Directory Service and Amazon Route 53 or How to Set Up DNS Resolution Between On-Premises Networks and AWS Using AWS Directory Service and Microsoft Active Directory.

AWS Identity and Access Management (IAM)

The following steps to establish the AWS Identity and Access Management (IAM) role and set up the IAM user for permissions are typically performed by an IAM administrator for your organization.

One-time creation of IAM Role

On the Choose Application page of Launch Wizard, under Permissions, Launch Wizard displays the IAM role required for the Amazon EC2 instances created by Launch Wizard to access other AWS services on your behalf. When you select Next, Launch Wizard attempts to discover the IAM role in your account. If the role exists, it is attached to the instance profile for the EC2 instances that Launch Wizard will launch into your account. If the role does not exist, Launch Wizard attempts to create the role with the same name, AmazonEC2RoleForLaunchWizard. This role is comprised of two IAM managed policies: AmazonSSMManagedInstanceCore and AmazonEC2RolePolicyForLaunchWizard. After the role is created, the IAM administrator can delegate the application deployment process to another IAM user who, in turn, must have the Launch Wizard IAM managed policy described in the following section.

IAM user setup

To deploy a SQL Server Always On application with Launch Wizard, you must create an Identity and Access Management (IAM) policy and attach it to your IAM user identity. The IAM policy defines the user permissions. If you do not already have an IAM user in your account, follow the steps listed in Create an IAM User in Your AWS Account.

When you have an IAM user in your account, create an IAM policy.

2. Choose Users from the left navigation pane.
3. Select the User name of the user to which you want to attach the policy.
4. Select Add permissions.
5. Select Attach existing policies directly.
6. Search for the policy named AmazonLaunchWizard_Fullaccess and select the check box to the left of the policy name.
7. Select Next: Review.
8. Verify that the correct policy is listed, and then select Add permissions.

Important

Log in with the user associated with the above policy when you use Launch Wizard.

AWS Secrets Manager

Launch Wizard uses AWS Secrets Manager to manage your domain and SQL Server account passwords. Your username and password is stored in Secrets Manager and is retrieved during the build process. The following resource policy is added to the secret so that the Launch Wizard IAM role can retrieve the secret. For more information about Secrets Manager, see the AWS Secrets Manager User Guide.

```json
{
    "Version" : "2012-10-17",
    "Statement" : [
        
```
Requirements for using custom AMIs (deployment on Windows)

We recommend that you use Amazon Windows license-included AMIs whenever possible. There are scenarios for which you may want to use a custom Windows AMI. For example, you may have existing licenses (BYOL), or you may have made changes to one of our public images and re-imaged it.

If you use Amazon Windows license-included AMIs, you are not required to perform any pre-checks on the AMI to ensure that it meets Launch Wizard requirements.

Launch Wizard relies on user data to begin the process of configuring SQL Server or RGW instances to launch in your account. For more information, see User Data Scripts. By default, all AWS Windows AMIs have user data execution enabled for the initial launch. To ensure that your custom AMIs are set up to run the User Data script at launch, follow the AWS recommended method to prepare your AMIs using EC2Launch v2. For more information about how to prepare your custom AMI using the options to Shutdown with Sysprep or Shutdown without Sysprep, see Create a Standard Amazon Machine Image Using Sysprep or EC2Launch v2 and Sysprep. If you want to directly enable user data as part of the custom AMI creation process, follow the steps for Subsequent Reboots or Starts under Running Commands on Your Windows Instance at Launch.

If you use a custom Windows AMI, the volume drive letter for the root partition should be C: because EC2Launch v2 and EC2Config rely on this configuration to install the components.

While not exhaustive, the following requirements cover most of the configurations whose alteration might impact the successful deployment of a SQL Server Always On application using Launch Wizard.

Support matrix

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server 2016</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>SQL Server 2017</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>SQL Server 2019</td>
<td>Currently not supported.</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

OS and SQL requirements

- Microsoft Windows Server 2012 R2 (Datacenter) (64-bit only)
- Microsoft Windows Server 2016 (Datacenter) (64-bit only)
- Microsoft Windows Server 2019 (Datacenter) (64-bit only)
- MBR-partitioned volumes and GUID Partition Table (GPT) partitioned volumes that are formatted using the NTFS file system
- English language pack only
• SQL Server Enterprise Edition 2019 or Standard Edition 2019
• The root volume drive for the custom AMI should be C:
• SQL Server is installed on the root drive

AWS software and drivers
• EC2Launch v2 (supported AMIs)
• EC2Config service (Windows Server 2012 R2)
• EC2Launch (Windows Server 2016)
• AWS SSM (SSM agent must be installed)
• AWS Tools for Windows PowerShell
• Network drivers (SRIOV, ENA)
• Storage drivers (NVMe, AWS PV)

Requirements for using custom AMIs (deployment on Linux)
There are occasions when you may want to use a custom Linux AMI. For example, you may have existing licenses (BYOL), or you may have made changes to one of our public images and re-imaged it.

If you use a custom Linux AMI, you must adhere to the following requirements:

• The operating system must be Ubuntu version 18.04 LTS.
• The system installer and administrator must be a sudo user and be able to log in to the cluster nodes using SSH.
• SQL Server for Linux must be a default installation.
• The SQL Server for Linux version must be 2019.
• The latest Microsoft SQL tools must be installed.

Requirements for using Windows license-included AMIs (SQL Server FCI configuration only)
You can use Windows license-included (LI) AMIs with SQL Bring-Your-Own-License (BYOL). Your SQL media must meet the following requirements to use Windows LI AMIs with SQL BYOL. Linux AMIs are not supported.

The SQL media must be:

• An ISO file.
• Hosted in an Amazon S3 bucket prefixed with LaunchWizard-*. 
• Included in a folder within the Amazon S3 bucket.
• Included in a public folder so that Launch Wizard can download and install the media.

Requirements for using Amazon FSx
Launch Wizard uses continuously available Amazon FSx file shares to host clustered databases. The Amazon FSx file shares are accessible from within an instance joined to the domain. You can either create a new Active Directory or connect to an existing Active Directory (managed or on-premises). If you connect to an existing Active Directory, you can use preexisting security groups. The security groups must satisfy port and security requirements for FSx to communicate with the domain, as described in
Using Amazon FSx with your self-managed Microsoft Active Directory and Using Amazon FSx with AWS Directory Service for Microsoft Active Directory.

If you are using an existing AWS Managed Active Directory instance, you must specify the ID of the managed Active Directory instance for FSx to be able to join the domain. The account must have the same access rights in the domain as described in Using Amazon FSx with your self-managed Microsoft Active Directory and Using Amazon FSx with AWS Directory Service for Microsoft Active Directory.

**Backup schedule**

Launch Wizard uses FSx defaults for setting up the backup schedule. You can change the default settings in the FSx console after the build completes.

The `WeeklyMaintenanceStartTime` follows the format of the `day:time`, where Monday is indicated by 1. The maintenance start time is set to begin on Saturday at 10pm.

```
WeeklyMaintenanceStartTime: '6:22:00'
DailyAutomaticBackupStartTime: '01:00'
AutomaticBackupRetentionDays: 7
```

**Configuration settings (deployment on Windows)**

The following configuration settings are applied when deploying a SQL Server Always On application with Launch Wizard.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Applies to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current EC2Launch v2 and SSM Agent</td>
<td>Supported AMIs</td>
</tr>
<tr>
<td>Current EC2Config and SSM Agent</td>
<td>Windows Server 2012 R2</td>
</tr>
<tr>
<td>Current EC2Launch and SSM Agent</td>
<td>Windows Server 2019 and 2016</td>
</tr>
<tr>
<td>Current AWS PV, ENA, and NVMe drivers</td>
<td>Windows Server 2019, 2016, and 2012 R2</td>
</tr>
<tr>
<td>Current SRIOV drivers</td>
<td>Windows Server 2019, 2016, and 2012 R2</td>
</tr>
<tr>
<td>Latest service pack</td>
<td></td>
</tr>
<tr>
<td>SQL Service configured to start automatically</td>
<td></td>
</tr>
<tr>
<td>SQL Service running</td>
<td></td>
</tr>
<tr>
<td><code>BUILTIN\Administrators</code> added to the SysAdmin server role</td>
<td></td>
</tr>
<tr>
<td>TCP port 1433 and UDP port 1434 open</td>
<td></td>
</tr>
<tr>
<td>Allow ICMP traffic through the firewall</td>
<td>Windows Server 2019, 2016, and 2012 R2</td>
</tr>
<tr>
<td>Allow RDP traffic through host firewall</td>
<td>Windows Server 2019, 2016, and 2012 R2</td>
</tr>
<tr>
<td>Enable file and printer sharing</td>
<td>Windows Server 2012 R2</td>
</tr>
<tr>
<td><code>RealTimeIsUniversal</code> registry key set</td>
<td>Windows Server 2019, 2016, and 2012 R2</td>
</tr>
<tr>
<td>SQL Server FCI</td>
<td>Windows Server 2019 and 2016</td>
</tr>
</tbody>
</table>
The following AMI settings can impact the Launch Wizard deployment:

**System Time**

*RealTimelsUniversal.* If disabled, Windows system time drifts when the time zone is set to a value other than UTC.

**Windows Firewall**

In most cases, Launch Wizard configures the correct protocols and ports. However, custom Windows Firewall rules could impact the cluster service. To ensure that your custom AMI works with Launch Wizard, see Service overview and network port requirements for Windows.

**Remote Desktop**

*Service Start.* Remote Desktop service must be enabled.

*Remote Desktop Connections.* Must be enabled.

**EC2Config (Server 2012 R2)**

*Installation.* We recommend using the latest version of EC2Config.

*Service Start.* EC2Config service should be enabled.

**Network Interface**

*DHCP Service Startup.* DHCP service should be enabled.

*DHCP on Ethernet.* DHCP should be enabled.

**Microsoft SQL Server**

*TCPIP.* Must be enabled for protocols in SQL Configuration Manager.

**PowerShell**

*Execution Policy.* The execution policy in all AWS license-included AMIs is set to Unrestricted. We recommend that you set this policy to Unrestricted when you set up SQL Server Always On Availability Groups using Launch Wizard. You can change the policy when setup is complete.

---

**Deploy an application with AWS Launch Wizard for SQL Server on Windows**

**Access AWS Launch Wizard**

You can launch AWS Launch Wizard from the AWS Launch Wizard console.

**Deploy AWS Launch Wizard on Windows**

**Topics**

- Deploy SQL Server Always On application (p. 173)
- Deploy SQL Failover Clustering application (p. 179)

**Deploy SQL Server Always On application**

The following steps guide you through a SQL Server Always On application deployment with AWS Launch Wizard after you have launched it from the console.
1. When you select **Choose application** from the AWS Launch Wizard landing page, you are directed to the **Choose application** wizard, where you are prompted to select the type of application that you want to deploy. Select **Microsoft SQL Server**, then **Create deployment**.

2. Under **Review Permissions**, Launch Wizard displays the AWS Identity and Access Management (IAM) role required for Launch Wizard to access other AWS services on your behalf. For more information about setting up IAM for Launch Wizard, see AWS Identity and Access Management (IAM) (p. 169). Choose **Next**.

3. On the **Configure application settings** page, select the **Operating System** on which you want to install SQL Server — in this case, **Windows**.

4. **Deployment model**. Choose **High availability deployment** to deploy your SQL Server Always On application across multiple Availability Zones or **Single instance deployment** to deploy your SQL Server application on a single node.

5. You are prompted to enter the specifications for the new deployment. The following tabs provide information about the specification fields.

**General**

- **Deployment name**. Enter a unique application name for your deployment.
- **Simple Notification Service (SNS) topic ARN** — **optional**. Specify an SNS topic where AWS Launch Wizard can send notifications and alerts. For more information, see the Amazon Simple Notification Service Developer Guide.
- **CloudWatch application monitoring (optional for HA deployments)**. Select the check box to set up monitors and automated insights for your deployment using CloudWatch Application Insights. For more information, see the Amazon CloudWatch User Guide.
- **Enable rollback on failed deployment**. By default, if a deployment fails, your provisioned resources will not be rolled back/deleted. This default configuration helps you to troubleshoot errors at the resource level as you debug deployment issues. If you want your provisioned resources to be immediately deleted if a deployment fails, select the check box.

**Connectivity**

Enter specifications for how you want to connect to your instance and configure your Virtual Private Cloud (VPC).

**Key pair name**

- Select an existing key pair from the dropdown list or create a new one. If you select **Create new key pair name**, you are directed to the Amazon EC2 console. From there, under **Network and Security**, choose **Key Pairs**. Choose **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**.

  **Important**

  This is the only opportunity for you to save the private key file. Download it and save it in a safe place. You must provide the name of your key pair when you launch an instance and provide the corresponding private key each time that you connect to the instance.

  Return to the Launch Wizard console and choose the refresh button next to the **Key Pairs** dropdown list. The newly created key pair appears in the dropdown list. For more information about key pairs, see Amazon EC2 Key Pairs and Windows Instances.

**Tenancy model (HA deployments only)**

Select your preferred tenancy. Each instance that you launch into a VPC has a tenancy attribute. The **Shared** tenancy option means that the instance runs on shared hardware. The **Dedicated**
Host (HA deployments) tenancy option means that the instance runs on a Dedicated Host, which is an isolated server with configurations that you can control. For more information, see Dedicated Hosts.

Virtual Private Cloud (VPC). Choose whether you want to use an existing VPC or create a new VPC.

- Select Virtual Private Cloud (VPC) option. Choose the VPC that you want to use from the dropdown list. If you choose to enable Remote Desktop Gateway access on single-node deployments, then your VPC must include one public subnet and one private subnet. It must include at least two private subnets for HA deployments. Your VPC must be associated with a DHCP Options Set to enable DNS translations to work. The private subnets must have outbound connectivity to the internet and other AWS services (S3, CFN, SSM, Logs). We recommend that you enable this connectivity with a NAT Gateway. For more information about NAT Gateways, see NAT Gateways in the Amazon VPC User Guide.

- Public Subnet. If you choose to enable Remote Desktop Gateway access on single-node deployments, then your VPC must include one public subnet and one private subnet. It must include at least two private subnets for HA deployments. Choose a public subnet for your VPC from the dropdown list. To continue, you must select the check box that indicates that the public subnet has been set up and the private subnets have outbound connectivity enabled.

To add a new public subnet

If a subnet’s traffic is routed to an internet gateway, the subnet is known as a public subnet. If, however, a subnet doesn’t have a route to the internet gateway, the subnet is known as a private subnet. To use an existing VPC that does not have a public subnet, you can add a new public subnet using the following steps.

- Follow the steps in Creating a Subnet in the Amazon VPC User Guide using the existing VPC you intend to use AWS Launch Wizard.

- To add an internet gateway to your VPC, follow the steps in Attaching an Internet Gateway in the Amazon VPC User Guide.

- To configure your subnets to route internet traffic through the internet gateway, follow the steps in Creating a Custom Route Table in the Amazon VPC User Guide. Use IPv4 format (0.0.0.0/0) for Destination.

- The public subnet should have the “auto-assign public IPv4 address” setting enabled. To enable this setting, follow the steps in Modifying the Public IPv4 Addressing Attribute for Your Subnet in the Amazon VPC User Guide.

- Availability Zone (AZ) configuration. You must choose at least two Availability Zones for High Availability (HA) deployments and one Availability Zone for single-node deployments, with one private subnet for each zone that you select. For HA deployments, select the Availability Zones within which you want to deploy your primary and secondary SQL nodes. Depending on the number of secondary nodes that you plan to use to set up a SQL Server Always On deployment, you may have to specify a private subnet for each of them. Cross-Region replication is not supported.

To create a private subnet

If a subnet doesn’t have a route to an internet gateway, the subnet is known as a private subnet. To create a private subnet, you can use the following steps. We recommend that you enable the outbound connectivity for each of your selected private subnets using a NAT Gateway. To enable outbound connectivity from private subnets with public subnet, see the steps in Creating a NAT Gateway to create a NAT Gateway in your chosen public subnet. Then, follow the steps in Updating Your Route Table for each of your chosen private subnets.
• Follow the steps in Creating a Subnet in the Amazon VPC User Guide using the existing VPC you will use in AWS Launch Wizard.

• When you create a VPC, it includes a main route table by default. On the Route Tables page in the Amazon VPC console, you can view the main route table for a VPC by looking for Yes in the Main column. The main route table controls the routing for all subnets that are not explicitly associated with any other route table. If the main route table for your VPC has an outbound route to an internet gateway, then any subnet created using the previous step, by default, becomes a public subnet. To ensure the subnets are private, you may need to create separate route table(s) for your private subnets. These route tables must not contain any routes to an internet gateway. Alternatively, you can create a custom route table for your public subnet and remove the internet gateway entry from the main route table.

If you selected Dedicated host tenancy, you must select a Dedicated Host for each Availability Zone. If you have not allocated any dedicated hosts to your account, you can choose Create new dedicated host to do so from the EC2 console.

• Remote Desktop Gateway preferences (single-node deployments only). When you select Set up Remote Desktop Gateway, enter the public subnet into which to deploy the RDGW instance.

• Remote Desktop Gateway access — Optional. Select Custom IP from the dropdown list. Enter the CIDR block. If you do not specify any value for the Custom IP parameter, Launch Wizard does not set the inbound RDP access (Port 3389) from any IP. You can choose to do this later by modifying the security group settings via the Amazon EC2 console. See Adding a Rule for Inbound RDP Traffic to a Windows Instance for instructions on adding a rule that allows inbound RDP traffic to your RDGW instance.

• Create new Virtual Private Cloud (VPC) option. Launch Wizard creates your VPC. You can optionally enter a VPC name tag. If you selected Dedicated Host tenancy for high availability deployments, select a primary and secondary Dedicated Host. If you haven't allocated any Dedicated Hosts to your account, select Create a new dedicated host. You will be directed to the EC2 console to create the new host.

• Remote Desktop Gateway preferences (single-node deployments only). When you select Set up Remote Desktop Gateway, only the Remote Desktop Gateway access information will be taken from the VPC.

• Remote Desktop Gateway access — Optional. Select Custom IP from the dropdown list. Enter the CIDR block. If you do not specify any value for the Custom IP parameter, Launch Wizard does not set the inbound RDP access (Port 3389) from any IP. You can choose to do this later by modifying the security group settings via the Amazon EC2 console. See Adding a Rule for Inbound RDP Traffic to a Windows Instance for instructions on adding a rule that allows inbound RDP traffic to your RDGW instance.

Active Directory

You can connect to an existing Active Directory or, for high availability deployments, you can create a new one. If you selected the Create new Virtual Private Cloud (VPC) option for high availability deployments, you must select Create a new Active Directory.

Connecting to existing AWS Managed Active Directory or On Premises Active Directory

From the dropdown list, select whether you want to use AWS Managed Active Directory, or On Premises Active Directory. If you select On Premises Active Directory, select the check box to verify that you have ensured a connection between the Active Directory and the VPC.
Follow the steps for granting permissions in the Active Directory Default Organizational Unit (OU).

- **Domain user name and password.** Enter the user name and password for your directory. For required permissions for the domain user, see Active Directory (Windows deployment) (p. 167). Launch Wizard stores the password in the Systems Manager Parameter Store of your account as a secure string parameter. It does not store the password on the service side. To create a functional SQL Server Always On deployment, it reads from the Parameter Store.

- **DNS address.** Enter the IP address of the DNS servers to which you are connecting. These servers must be reachable from within the VPC that you selected.

- **Optional DNS address.** If you would like to use a backup DNS server, enter the IP address of the DNS server that you want to use as backup. These servers must be reachable from within the VPC that you selected.

- **Domain DNS name.** Enter the Fully Qualified Domain Name (FQDN) of the forest root domain used for the Active Directory. When you choose to create a new Active Directory, Launch Wizard creates a domain admin user on your Active Directory.

**Creating a new AWS Managed Active Directory through Launch Wizard**

- **Domain user name and password.** The domain user name is preset to “admin.” Enter a password for your directory. Launch Wizard stores the password in the Systems Manager Parameter Store of your account as a secure string parameter. It does not store the password on the server side. To create a functional SQL Server Always On deployment, it reads from the Parameter Store.

- **Domain DNS name.** Enter a Fully Qualified Domain Name (FQDN) of the forest root domain used for the Active Directory. When you choose to create a new Active Directory, Launch Wizard creates a domain admin user on your Active Directory.

**Creating an on-premises Active Directory through Launch Wizard**

Launch Wizard allows you to connect to your on-premises environment with AWS Direct Connect.

**SQL Server**

When you use an existing Active Directory, you have the option of using an existing SQL Server service account or creating a new account. If you create a new Active Directory account, you must create a new SQL Server account.

- **User name and password.** If you are using an existing SQL Server service account, provide your user name and password. This SQL Server service account should be part of the Managed Active Directory in which you are deploying. If you are creating a new SQL Server service account through Launch Wizard, enter a user name for the SQL Server service account. Create a complex Password that is at least 8 characters long, and then reenter the password to verify it. See Password Policy for more information.

- **SQL Server install type.** Select the version of SQL Server Enterprise that you want to deploy. You can select an AMI from either the License-included AMI or Custom AMI dropdown lists.

- **Additional SQL Server settings (optional).** You can optionally specify additional nodes and their subnets.

  - **Nodes.** Enter a Primary SQL node name and a Secondary SQL node name (HA deployments only).

  - **Additional secondary SQL node (HA deployments only, maximum of 5).** Enter a secondary Node name, and select the Access type, the Private subnet, and the Dedicated
**host**, if applicable, for the additional secondary SQL node from the dropdown lists. You can add more secondary nodes by selecting **Add additional secondary node**.

- **Witness node (optional, HA deployments only)**. For improved fault tolerance, select the check box to add a file share quorum witness node.

- **Additional naming.** Enter a **Database name**. For HA deployments, enter an **Availability group name**, a **Listener name**, and a **Cluster name**.

6. When you are satisfied with your configuration selections, select **Next**. If you don't want to complete the configuration, select **Cancel**. When you select **Cancel**, all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select **Previous**.

7. After configuring your application, you are prompted to define the infrastructure requirements for the new deployment on the **Define infrastructure requirements** page. The following tabs provide information about the input fields.

**Storage and Compute**

### Infrastructure requirements based on infrastructure

You can choose to select your instances and volume types, or to use AWS recommended resources. If you choose to use AWS recommended resources, you have the option of defining your high availability cluster needs. If no selections are made, default values are assigned.

- **Number of instance cores.** Choose the number of CPU cores for your infrastructure. The default value assigned is 4.

- **Network performance.** Choose your preferred network performance in Gbps.

- **Memory (GB).** Choose the amount of RAM that you want to attach to your EC2 instances. The default value assigned is 4 GB.

- **Type of storage drive.** Select the storage drive type for the SQL data and tempdb volumes. The default value assigned is SSD.

- **SQL Server throughput.** Select the sustained SQL Server throughput that you need.

- **Recommended resources.** Launch Wizard displays the system-recommended resources based on your infrastructure selections. If you want to change the recommended resources, select different infrastructure requirements.

### Infrastructure requirements based on instance type

You can choose to select your instance and volume type, or to use AWS recommended resources. If no selections are made, default values are assigned.

- **Instance type.** Select your preferred instance type from the dropdown list.

- **Volume type.** Choose your preferred EBS volume type. For more information about volume types, see **Amazon EBS volume types**.

### Drive letters and volume size

- **Drive letter.** Select the storage drive letter for **Root drive, Logs, Data, and Backup** volumes.

  **Important**

  For custom AMIs, Launch Wizard assumes the root volume drive is C:.

- **Volume size.** Select the size of the SQL Server data volume in Gb for **Root drive, Logs, Data, and Backup** volumes. SQL Server logs and data will be staged on the same data volume for this deployment. Make sure that you select an adequate size for the data volume.
Tags-Optional

You can provide optional custom tags for the resources Launch Wizard creates on your behalf. For example, you can set different tags for EC2 instances, EBS volumes, VPC, and subnets. If you select All, you can assign a common set of tags to your resources. Launch Wizard assigns tags with a fixed key LaunchWizardResourceGroupID and value that corresponds to the ID of the AWS resource group created for a deployment. Launch Wizard does not support custom tagging for root volumes.

Estimated on-demand cost to deploy additional resources

AWS Launch Wizard provides an estimate for application charges incurred to deploy the selected resources. The estimate updates each time you change a resource type in the Wizard. The provided estimates are only for general comparisons. They are based upon On-Demand costs and your actual costs may be lower.

8. When you are satisfied with your infrastructure selections, select Next. If you don't want to complete the configuration, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select Previous.

9. On the Review and deploy page, review your configuration details. If you want to make changes, select Previous. To stop, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the landing page. When you choose Deploy, you agree to the terms of the Acknowledgment.

10. Launch Wizard validates the inputs and notifies you of any issues you must address.

11. When validation is complete, Launch Wizard deploys your AWS resources and configures your SQL Server Always On application. Launch Wizard provides you with status updates about the progress of the deployment on the Deployments page. From the Deployments page, you can view the list of current and previous deployments.

12. When your deployment is ready, a notification informs you that your SQL Server application is successfully deployed. If you have set up an SNS notification, you are also alerted through SNS. You can manage and access all of the resources related to your SQL Server Always On application by selecting the deployment, and then selecting Manage from the Actions dropdown list.

13. When the SQL Server Always On application is deployed, you can access your Amazon EC2 instances through the EC2 console. You can also use AWS SSM to manage your SQL Server Always On application for future updates and patches through built-in integration via resource groups.

Deploy SQL Failover Clustering application

The following steps guide you through a SQL Failover Clustering application deployment with AWS Launch Wizard after you have launched it from the console.

1. When you select Choose application from the AWS Launch Wizard landing page, you are directed to the Choose application wizard, where you are prompted to select the type of application that you want to deploy. Select Microsoft SQL Server, then Create deployment.

2. Under Review Permissions, Launch Wizard displays the AWS Identity and Access Management (IAM) role required for Launch Wizard to access other AWS services on your behalf. For more information about setting up IAM for Launch Wizard, see AWS Identity and Access Management (IAM) (p. 169). Choose Next.

3. On the Configure application settings page, select the Operating System on which you want to install SQL Server — in this case, Windows.

4. Deployment model. Choose High availability deployment, and then choose Always On Failover Cluster Instances to deploy a SQL Server Failover Clustering (FCI) application across multiple Availability Zones.
5. You are prompted to enter the specifications for the new deployment. The following tabs provide information about the specification fields.

General

- **Deployment name.** Enter a unique application name for your deployment.
- **Simple Notification Service (SNS) topic ARN — optional.** Specify an SNS topic where AWS Launch Wizard can send notifications and alerts. For more information, see the Amazon Simple Notification Service Developer Guide.
- **CloudWatch application monitoring (optional for HA deployments).** Select the check box to set up monitors and automated insights for your deployment using CloudWatch Application Insights. For more information, see the Amazon CloudWatch User Guide.
- **Enable rollback on failed deployment.** By default, if a deployment fails, your provisioned resources will not be rolled back/deleted. This default configuration helps you to troubleshoot errors at the resource level as you debug deployment issues. If you want your provisioned resources to be immediately deleted if a deployment fails, select the check box.

Connectivity

Enter the specifications for how you want to connect to your instance and configure your Virtual Private Cloud (VPC).

**Key pair name**

- Select an existing key pair from the dropdown list or create a new one. If you select Create new key pair name, you are directed to the Amazon EC2 console. From there, under Network and Security, choose Key Pairs. Choose Create a new key pair, enter a name for the key pair, and then choose Download Key Pair.

  **Important**
  
  This is the only opportunity for you to save the private key file. Download it and save it in a safe place. You must provide the name of your key pair when you launch an instance and provide the corresponding private key each time that you connect to the instance.

  Return to the Launch Wizard console and choose the refresh button next to the Key Pairs dropdown list. The newly created key pair appears in the dropdown list. For more information about key pairs, see Amazon EC2 Key Pairs and Windows Instances.

**Tenancy model (HA deployments only)**

Select your preferred tenancy. Each instance that you launch into a VPC has a tenancy attribute. The **Shared** tenancy option means that the instance runs on shared hardware. The **Dedicated Host (HA deployments)** tenancy option means that the instance runs on a Dedicated Host, which is an isolated server with configurations that you can control. For FCI deployments, select Shared tenancy.

**Virtual Private Cloud (VPC).** Choose whether you want to use an existing VPC or create a new VPC.

- **Select Virtual Private Cloud (VPC) option.** Choose the VPC that you want to use from the dropdown list. If you choose to enable Remote Desktop Gateway access, then your VPC must include at least one public subnet and two private subnets for HA deployments. Your VPC must be associated with a DHCP Options Set to enable DNS translations to work. The private subnets must have outbound connectivity to the internet and other AWS services (S3, CFN, SSM, Logs). We recommend that you enable this connectivity with a NAT Gateway. For more information about NAT Gateways, see NAT Gateways in the Amazon VPC User Guide.
• **Public Subnet.** If you choose to enable Remote Desktop Gateway access, then your VPC must include at least one public subnet and two private subnets for HA deployments. Choose a public subnet for your VPC from the dropdown list. To continue, you must select the check box that indicates that the public subnet has been set up and the private subnets have outbound connectivity enabled.

**To add a new public subnet**

If a subnet’s traffic is routed to an internet gateway, the subnet is known as a public subnet. If, however, a subnet doesn’t have a route to the internet gateway, the subnet is known as a private subnet. To use an existing VPC that does not have a public subnet, you can add a new public subnet using the following steps.

- Follow the steps in [Creating a Subnet in the Amazon VPC User Guide](#) using the existing VPC you intend to use AWS Launch Wizard.
- To add an internet gateway to your VPC, follow the steps in [Attaching an Internet Gateway](#) in the Amazon VPC User Guide.
- To configure your subnets to route internet traffic through the internet gateway, follow the steps in [Creating a Custom Route Table](#) in the Amazon VPC User Guide. Use IPv4 format (0.0.0.0/0) for Destination.
- The public subnet should have the “auto-assign public IPv4 address” setting enabled. To enable this setting, follow the steps in [Modifying the Public IPv4 Addressing Attribute for Your Subnet](#) in the Amazon VPC User Guide.

• **Availability Zone (AZ) configuration.** You must choose at least two Availability Zones for High Availability (HA) deployments, with one private subnet for each zone that you select. For HA deployments, select the **Availability Zones** within which you want to deploy your primary and secondary SQL nodes. Depending on the number of secondary nodes that you plan to use to set up a SQL Server Always On deployment, you may have to specify a private subnet for each of them. Cross-Region replication is not supported.

**To create a private subnet**

If a subnet doesn’t have a route to an internet gateway, the subnet is known as a private subnet. To create a private subnet, you can use the following steps. We recommend that you enable the outbound connectivity for each of your selected private subnets using a NAT Gateway. To enable outbound connectivity from private subnets with public subnet, see the steps in [Creating a NAT Gateway](#) to create a NAT Gateway in your chosen public subnet. Then, follow the steps in [Updating Your Route Table](#) for each of your chosen private subnets.

- Follow the steps in [Creating a Subnet](#) in the Amazon VPC User Guide using the existing VPC you will use in AWS Launch Wizard.
- When you create a VPC, it includes a main route table by default. On the Route Tables page in the Amazon VPC console, you can view the main route table for a VPC by looking for Yes in the Main column. The main route table controls the routing for all subnets that are not explicitly associated with any other route table. If the main route table for your VPC has an outbound route to an internet gateway, then any subnet created using the previous step, by default, becomes a public subnet. To ensure the subnets are private, you may need to create separate route table(s) for your private subnets. These route tables must not contain any routes to an internet gateway. Alternatively, you can create a custom route table for your public subnet and remove the internet gateway entry from the main route table.

- **Remote Desktop Gateway preferences.** When you select Set up Remote Desktop Gateway, enter the public subnet into which to deploy the RDGW instance.
- **Remote Desktop Gateway access — Optional.** Select Custom IP from the dropdown list. Enter the CIDR block. If you do not specify any value for the Custom IP parameter, Launch Wizard does not set the inbound RDP access (Port 3389) from any IP. You can choose to do...
Deploy on Windows

this later by modifying the security group settings via the Amazon EC2 console. See Adding a Rule for Inbound RDP Traffic to a Windows Instance for instructions on adding a rule that allows inbound RDP traffic to your RDGW instance.

- **Create new Virtual Private Cloud (VPC) option.** Launch Wizard creates your VPC. You can optionally enter a **VPC name tag**.
  
  - **Remote Desktop Gateway preferences.** When you select Set up Remote Desktop Gateway, only the Remote Desktop Gateway access information will be taken from the VPC.
  
  - **Remote Desktop Gateway access — Optional.** Select Custom IP from the dropdown list. Enter the CIDR block. If you do not specify any value for the Custom IP parameter, Launch Wizard does not set the inbound RDP access (Port 3389) from any IP. You can choose to do this later by modifying the security group settings via the Amazon EC2 Console. See Adding a Rule for Inbound RDP Traffic to a Windows Instance for instructions on adding a rule that allows inbound RDP traffic to your RDGW instance.

**Active Directory**

You can connect to an existing Active Directory or create a new one. If you selected the **Create new Virtual Private Cloud (VPC) option** for high availability deployments, you must select Create a new Active Directory.

**Connecting to existing AWS Managed Active Directory or On Premises Active Directory**

From the dropdown list, select whether you want to use AWS Managed Active Directory, or On Premises Active Directory. If you select On Premises Active Directory, select the check box to verify that you have ensured a connection between the Active Directory and the VPC.

Follow the steps for granting permissions in the Active Directory Default Organizational Unit (OU).

- **Domain user name and password.** Enter the user name and password for your directory. For required permissions for the domain user, see Active Directory (Windows deployment) (p. 167). Launch Wizard stores the password in the Systems Manager Parameter Store of your account as a secure string parameter. It does not store the password on the service side. To create a functional SQL Server FCI deployment, Launch Wizard reads from the Parameter Store.
  
  - **DNS address.** Enter the IP address of the DNS servers to which you are connecting. These servers must be reachable from within the VPC that you selected.
  
  - **Optional DNS address.** If you would like to use a backup DNS server, enter the IP address of the DNS server that you want to use as backup. These servers must be reachable from within the VPC that you selected.
  
  - **Domain DNS name.** Enter the Fully Qualified Domain Name (FQDN) of the forest root domain used for the Active Directory. When you choose to create a new Active Directory, Launch Wizard creates a domain admin user on your Active Directory.
  
  - **Domain User security group — optional.** To specify an existing security group, select one from the dropdown list. The prerequisites for adding security groups can be viewed by selecting Info.

**Creating a new AWS Managed Active Directory through Launch Wizard**

- **Domain user name and password.** The domain user name is preset to "admin." Enter a password for your directory. Launch Wizard stores the password in the Systems Manager Parameter Store of your account as a secure string parameter. It does not store the password
on the server side. To create a functional SQL Server FCI deployment, Launch Wizard reads from the Parameter Store.

- **Domain DNS name.** Enter a Fully Qualified Domain Name (FQDN) of the forest root domain used for the Active Directory. When you choose to create a new Active Directory, Launch Wizard creates a domain admin user on your Active Directory.

**Creating an on-premises Active Directory through Launch Wizard**

Launch Wizard allows you to connect to your on-premises environment with AWS Direct Connect.

**SQL Server**

When you use an existing Active Directory, you have the option of using an existing SQL Server service account or creating a new account. If you create a new Active Directory account, you must create a new SQL Server account.

- **User name and password.** If you are using an existing SQL Server service account, provide your user name and password. This SQL Server service account should be part of the Managed Active Directory in which you are deploying. If you are creating a new SQL Server service account through Launch Wizard, enter a user name for the SQL Server service account. Create a complex Password that is at least 8 characters long, and then reenter the password to verify it. See Password Policy for more information.

- **SQL Server install type.** Select the version of SQL Server Enterprise that you want to deploy. You can select an AMI from either the License-included AMI or Custom AMI dropdown lists.

- **Additional SQL Server settings (optional).** You can optionally specify additional nodes and their subnets.
  - **Nodes.** Enter a **Primary SQL node name** and a **Secondary SQL node name**.
  - **Additional naming.** You can specify a name for your SQL FCI instance and your Cluster.
  - **Additional secondary SQL node (maximum of 5).** Enter a secondary **Node name**, and select the **Access type**, the **Private subnet**, and the **Dedicated host**, if applicable, for the additional secondary SQL node from the dropdown lists. You can add more secondary nodes by selecting **Add additional secondary node**.
  - **Witness node (optional).** For improved fault tolerance, select the check box to add a file share quorum witness node.
  - **Additional naming.** Enter a **Database name**, an **Availability group name**, a **Listener name**, and a **Cluster name**.

6. When you are satisfied with your configuration selections, select **Next**. If you don't want to complete the configuration, select **Cancel**. When you select **Cancel**, all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select **Previous**.

7. After configuring your application, you are prompted to define the infrastructure requirements for the new deployment on the **Define infrastructure requirements** page. The following tabs provide information about the input fields.

**Storage and Compute**

**Infrastructure requirements based on infrastructure**

You can choose to select your instances and volume types, or to use AWS recommended resources. If you choose to use AWS recommended resources, you have the option of defining your high availability cluster needs. If no selections are made, default values are assigned.

**Instances**
- **Cores.** Choose the number of CPU cores for your infrastructure. The default value assigned is 4.
- **Network performance.** Choose your preferred network performance in Gbps.
- **Memory (GB).** Choose the amount of RAM that you want to attach to your EC2 instances. The default value assigned is 4 GB.

**Storage and performance**
- **Type of storage drive.** The default value assigned is SSD for FCI application deployments.
- **Average and peak IOPS.** Select the average and peak IOPS required for your FSx share.
- **Allocated storage space.** Select the amount of storage required for your FSx drive.
- **Recommended resources.** Launch Wizard displays the system-recommended resources based on your infrastructure selections. If you want to change the recommended resources, select different infrastructure requirements.

**Infrastructure requirements based on instance type**
You can choose to select your instance and storage capacity, or to use AWS recommended resources. If no selections are made, default values are assigned.
- **Instance type.** Select your preferred instance type from the dropdown list.
- **Storage capacity.** Choose your preferred EBS volume type. For more information about volume types, see Amazon EBS volume types.
- **Throughput capacity.** Select the required sustained SQL Server throughput.

**Drive letters and volume size**
- **Drive letter.** Select the storage drive letter for **Root drive, Logs, Data,** and **Backup** volumes.
  
  **Important**
  For custom AMIs, Launch Wizard assumes the root volume drive is C:.

- **Volume size.** Select the size of the SQL Server data volume in Gb for **Root drive, Logs, Data,** and **Backup** volumes. SQL Server logs and data will be staged on the same data volume for this deployment. Make sure that you select an adequate size for the data volume.

**Tags-Optional**
You can provide optional custom tags for the resources Launch Wizard creates on your behalf. For example, you can set different tags for EC2 instances, EBS volumes, VPC, and subnets. If you select **All,** you can assign a common set of tags to your resources. Launch Wizard assigns tags with a fixed key LaunchWizardResourceGroupID and value that corresponds to the ID of the AWS resource group created for a deployment. Launch Wizard does not support custom tagging for root volumes.

**Estimated on-demand cost to deploy additional resources**
AWS Launch Wizard provides an estimate for application charges incurred to deploy the selected resources. The estimate updates each time you change a resource type in the Wizard. The provided estimates are only for general comparisons. They are based upon On-Demand costs and your actual costs may be lower.

8. **When you are satisfied with your infrastructure selections, select Next.** If you don't want to complete the configuration, select **Cancel.** When you select **Cancel,** all of the selections on the specification
page are lost and you are returned to the landing page. To go to the previous screen, select Previous.

9. On the Review and deploy page, review your configuration details. If you want to make changes, select Previous. To stop, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the landing page. When you choose Deploy, you agree to the terms of the Acknowledgment.

10. Launch Wizard validates the inputs and notifies you of any issues you must address.

11. When validation is complete, Launch Wizard deploys your AWS resources and configures your SQL Server FCI application. Launch Wizard provides you with status updates about the progress of the deployment on the Deployments page. From the Deployments page, you can view the list of current and previous deployments.

12. When your deployment is ready, a notification informs you that your SQL Server application is successfully deployed. If you have set up an SNS notification, you are also alerted through SNS. You can manage and access all of the resources related to your SQL Server FCI application by selecting the deployment, and then selecting Manage from the Actions dropdown list.

13. When the SQL Server FCI application is deployed, you can access your Amazon EC2 instances through the EC2 console. You can also use AWS SSM to manage your SQL Server FCI application for future updates and patches through built-in integration via resource groups.

Deploy an application with AWS Launch Wizard for SQL Server on Ubuntu

Topics
- Access AWS Launch Wizard (p. 185)
- Deploy AWS Launch Wizard on Ubuntu (p. 185)
- Post-deployment cluster tasks (p. 190)

Access AWS Launch Wizard

You can launch AWS Launch Wizard from the AWS Launch Wizard console.

Deploy AWS Launch Wizard on Ubuntu

The following steps guide you through a SQL Server application deployment with AWS Launch Wizard on the Ubuntu platform after you have launched it from the console. For SQL Server deployments on Ubuntu, you must use an instance type built on the Nitro System. EBS volumes are exposed as NVMe block devices on instances built with the Nitro System. Device names that are specified for NVMe EBS volumes in a block device mapping are renamed using NVMe device names (/dev/nvme[0-26].n1). Launch Wizard deployments on Ubuntu do not support block devices on Xen-virtualized instances.

1. When you select Choose application from the AWS Launch Wizard landing page, you are directed to the Choose application wizard, where you are prompted to select the type of application that you want to deploy. Select Microsoft SQL Server, then Create deployment.

2. Under Review Permissions, Launch Wizard displays the AWS Identity and Access Management (IAM) role required for Launch Wizard to access other AWS services on your behalf. For more information about setting up IAM for Launch Wizard, see AWS Identity and Access Management (IAM) (p. 169). Choose Next.

3. On the Configure application settings page, select the Operating System on which you want to install SQL Server — in this case, Ubuntu.
4. **Deployment model.** Choose **High availability deployment** to deploy your SQL Server Always On application across multiple Availability Zones or **Single instance deployment** to deploy your SQL Server application on a single node.

5. You are prompted to enter specifications for the new deployment. The following tabs provide information about the input fields.

**General**

- **Deployment name.** Enter a unique application name for your deployment.
- **Simple Notification Service (SNS) topic ARN (Optional).** Specify an SNS topic where AWS Launch Wizard can send notifications and alerts. For more information, see the *Amazon Simple Notification Service Developer Guide.*
- **Enable rollback on failed deployment.** By default, if a deployment fails, your provisioned resources will not be rolled back/deleted. This default configuration helps you to troubleshoot errors at the resource level as you debug deployment issues. If you want your provisioned resources to be immediately deleted if a deployment fails, select the check box.

**Connectivity**

Enter your requirements for how you want to connect to your application instance and what kind of Virtual Private Cloud (VPC) you want to set up.

**Key pair name**

- Select an existing key pair from the dropdown list or create a new one. If you select **Create new key pair name** to create a new key pair, you are directed to the Amazon EC2 console. From there, under **Network and Security**, choose **Key Pairs**. Choose **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**.

**Important**

This is your only opportunity to save the private key file. Download it and save it in a safe place. You must provide the name of your key pair when you launch an instance, and provide the corresponding private key each time that you connect to the instance.

Return to the Launch Wizard console and choose the refresh button next to the **Key Pairs** dropdown list. The newly created key pair appears in the dropdown list. For more information about key pairs, see *Amazon EC2 Key Pairs and Windows Instances.*

**Virtual Private Cloud (VPC).** Choose whether you want to use an existing VPC or create a new VPC.

- **Select Virtual Private Cloud (VPC) option.** Choose the VPC that you want to use from the dropdown list. Your VPC must contain one public subnet. For HA deployments, it must also contain, at least, three private subnets. For single node deployments, it must contain one private subnet. The private subnets must have outbound connectivity to the internet and other AWS services (S3, CFN, SSM, Logs). We recommend that you enable this connectivity with a NAT Gateway. For more information about NAT Gateways, see *NAT Gateways* in the Amazon VPC User Guide.
- **Public Subnet.** Your VPC must contain one public subnet. For HA deployments it must also contain three private subnets. For single node deployments, it must contain one private subnet. Choose a public subnet for your VPC from the dropdown list. To continue, you must select the check box that indicates that the public subnet has been set up and each of the selected private subnets have outbound connectivity enabled.
**To add a new public subnet**

If the traffic of a subnet is routed to an internet gateway, the subnet is known as a public subnet. If, however, a subnet doesn't have a route to the internet gateway, the subnet is known as a private subnet. To use an existing VPC that does not have a public subnet, you can add a new public subnet using the following steps.

- Follow the steps in [Creating a Subnet in the Amazon VPC User Guide](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-create-subnet-quick.html) using the existing VPC you intend to use AWS Launch Wizard.
- To add an internet gateway to your VPC, follow the steps in [Attaching an Internet Gateway](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-attach-internet-gateway.html) in the Amazon VPC User Guide.
- To configure your subnets to route internet traffic through the internet gateway, follow the steps in [Creating a Custom Route Table](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-create-route-table.html) in the Amazon VPC User Guide. Use IPv4 format (0.0.0.0/0) for Destination.
- The public subnet should have the "auto-assign public IPv4 address" setting enabled. To enable this setting, follow the steps in [Modifying the Public IPv4 Addressing Attribute for Your Subnet](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-change-public-assignable-range.html) in the Amazon VPC User Guide.
- **Availability Zone (AZ) configuration.** You must choose at least three Availability Zones for High Availability (HA) deployments and one Availability Zone for single-node deployments, with one private subnet for each Availability Zone that you select. From the dropdown lists, select the [Availability Zones](https://console.aws.amazon.com/vpc/home?region=us-east-1) within which you want to deploy your **primary**, **secondary**, and **configuration** nodes.

**To create a private subnet**

If a subnet doesn't have a route to an internet gateway, the subnet is known as a private subnet. To create a private subnet, perform the following steps. We recommend that you enable the outbound connectivity for each of your selected private subnets using a NAT Gateway. To enable outbound connectivity from private subnets to public subnets, see the steps in [Creating a NAT Gateway](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-choose-nat-gateway.html) to create a NAT Gateway in your chosen public subnet. Then, follow the steps in [Updating Your Route Table](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-update-route-table.html) for each of your chosen private subnets.

- Follow the steps in [Creating a Subnet](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-create-subnet-quick.html) in the Amazon VPC User Guide using the existing VPC you will use in AWS Launch Wizard.
- When you create a VPC, it includes a main route table by default. On the [Route Tables](https://console.aws.amazon.com/vpc/home?region=us-east-1) page in the Amazon VPC console, you can view the main route table for a VPC by looking for Yes in the **Main** column. The main route table controls the routing for all subnets that are not explicitly associated with any other route table. If the main route table for your VPC has an outbound route to an internet gateway, then any subnet created using the previous step, by default, becomes a public subnet. To ensure the subnets are private, you may need to create one separate route table for all of your private subnets. This route table must not contain any routes to an internet gateway. Verify that all of the private subnets have the same route table association.

- **Create new Virtual Private Cloud (VPC) option.** Launch Wizard creates your VPC. You can optionally enter a VPC name tag.

**SQL Server**

**SQL Server configuration**

- **User name and password.** By default, Launch Wizard applies the user name `sa`. This system administrator account is used for SQL Server management. Create a complex password that is
at least 8 characters long, and then reenter the password to verify it. See Password Policy for more information.

- **Floating IP Address (HA and existing VPC deployments only).** This field is available when you select a Virtual Private Cloud (VPC). The IP address that you enter is used as the endpoint for your SQL Server Availability Group listener. Launch Wizard creates a route from this IP address to the SQL primary node in your route table. Verify that the IP address is not already in use within your VPC and is outside of all of the provided subnet CIDRs.

- **Amazon Machine Image (AMI).** Select the version of Microsoft SQL Server Enterprise to deploy. You can select an AMI from the lists of either license-included or custom AMIs.

**Pacemaker cluster configuration (HA deployments only)**

Pacemaker is a high-availability cluster resource manager. This software runs on a set of hosts, or cluster of nodes, to preserve integrity and minimize the downtime of selected services or resources. Pacemaker is maintained by the ClusterLabs community.

- **Pacemaker cluster name.** Enter a name to identify your pacemaker cluster.
- **Pacemaker cluster username.** By default, Launch Wizard applies the pacemaker username hacluster. This username is used to securely communicate between cluster nodes.
- **Pacemaker cluster password.** Create a complex password that is at least 8 characters long, and then reenter the password to verify it. See Password Policy for more information.

**SQL - Pacemaker cluster connection settings (HA deployments only)**

After you configure Pacemaker cluster and SQL Server, you must create a user in SQL Server to communicate with Pacemaker.

- **SQL Pacemaker user name and password.** Enter a user name for SQL Server to communicate with the Pacemaker cluster. Create a complex password that is at least 8 characters long, and then reenter the password to verify it. See Password Policy for more information.
- **S3 location for node certificates.** An Amazon S3 bucket location is required by the SQL nodes to share self-signed certificates with each other. Provide the bucket or object locations and verify that the names begin with launchwizard-.

**Additional SQL Server settings (optional)**

- **Nodes.** Enter a Primary SQL node name, a Secondary SQL node name, and a Configuration node name.
- **Additional naming.** Enter a Database name and an Availability group name.

6. When you are satisfied with your configuration selections, select Next. If you don't want to complete the configuration, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select Previous.

7. After configuring your application, you are prompted to define the infrastructure requirements for the new deployment on the Define infrastructure requirements page. The following tabs provide information about the input fields.

**Infrastructure requirements based on infrastructure**

You can choose to select your instances, storage and performance, and volume types, or to use AWS recommended resources. If you choose to use AWS recommended resources, you have the
option of defining your high availability cluster needs. If no selections are made, default values are assigned.

- **Number of instance cores.** Choose the number of CPU cores for your infrastructure. The default value assigned is 4.
- **Network performance.** Choose your preferred network performance in Gbps.
- **Memory (GB).** Choose the amount of RAM that you want to attach to your EC2 instances. The default value assigned is 4 GB.
- **Type of storage drive.** Select the storage drive type for the SQL data and tempdb volumes. The default value assigned is SSD.
- **SQL Server throughput.** Select the sustained SQL Server throughput that you need.
- **Recommended resources.** Launch Wizard displays the system-recommended resources based on your infrastructure selections. If you want to change the recommended resources, select different infrastructure requirements.

**Infrastructure requirements based on instance type**

You can choose to select your instance and volume type, or to use AWS recommended resources. If no selections are made, default values are assigned.

- **Instance type.** Select your preferred instance type from the dropdown list.
- **Volume type.** Choose your preferred EBS volume type. For more information about volume types, see Amazon EBS volume types

**Volume sizes**

- **Volume size.** Select the size of the SQL Server data volume in Gb for Temporary database, Logs, Data, and Backup volumes. SQL Server logs and data will be staged on the same data volume for this deployment. Make sure that you select an adequate size for the data volume.

**Tags-Optional**

You can provide optional custom tags for the resources Launch Wizard creates on your behalf. For example, you can set different tags for EC2 instances, EBS volumes, VPC, and subnets. If you select All, you can assign a common set of tags to your resources. Launch Wizard assigns tags with a fixed key LaunchWizardResourceGroupID and value that corresponds to the ID of the AWS resource group created for a deployment. Launch Wizard does not support custom tagging for root volumes.

**Estimated on-demand cost to deploy additional resources**

AWS Launch Wizard provides an estimate for application charges incurred to deploy the selected resources. The estimate updates each time you change a resource type in the wizard. The provided estimates are for general comparisons only. They are based upon On-Demand costs and your actual costs may be lower.

8. When you are satisfied with your infrastructure selections, select Next. If you don't want to complete the configuration, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select Previous.

9. On the Review and deploy page, review your configuration details. If you want to make changes, select Previous. To stop, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the service landing page. When you choose Deploy, you agree to the terms of the Note at the bottom of the page.

10. Launch Wizard validates the inputs and notifies you if you must update a specification.
11. When validation is complete, Launch Wizard deploys your AWS resources and configures your SQL Server Always On application. Launch Wizard provides you with status updates about the progress of the deployment on the Deployments page. From the Deployments page, you can view the list of current and previous deployments.

12. When your deployment is ready, a notification informs you that your SQL Server application is successfully deployed. If you have set up an SNS notification, you are also alerted through SNS. You can manage and access all of the resources related to your SQL Server Always On application by selecting the deployment, and then selecting Manage from the Actions dropdown list.

13. When the SQL Server Always On application is deployed, you can access your Amazon EC2 instances through the EC2 console. You can also use AWS SSM to manage your SQL Server Always On application for future updates and patches through built-in integration via resource groups.

### Post-deployment cluster tasks

The Launch Wizard Pacemaker implementation includes three cluster nodes: primary, secondary, and configuration only. The primary node provides the Microsoft SQL Server for Ubuntu resource and the floating IP address. To ensure that the cluster operates correctly, some administrative tasks must be performed in a specific way. If these tasks are performed incorrectly, then Pacemaker may identify the activity as a resource failure and attempt to fail over the resources to the secondary node. If the resources are failed over to the secondary node, the cluster can remain in an unknown state, which can impact user access.

There are four primary tasks: **Start Cluster**, **Stop Cluster**, **Move Resources**, and **Recovery**. These tasks must be carried out by a sudo user with an SSH connection to any of the cluster nodes. Before performing any of these tasks, verify the cluster status using `pcs resource status --all`. This command returns all cluster issues. All issues must be addressed prior to performing any administrative tasks.

#### Start cluster

1. Log in to a cluster node using a sudo user over an SSH connection.
2. Verify that all cluster nodes are available.
3. Verify cluster status using the following command: `pcs resource --all`.
   
   Address all issues before attempting to start the cluster.
4. Start all cluster nodes using the following command: `pcs cluster start --all --wait`.
5. Verify that the cluster has started using the following command:`pcs resource --all`.

   The output provides information about the cluster nodes and cluster resources. All cluster nodes should be online and all resource agents should be visible and allocated to their assigned cluster nodes.
6. Verify that the availability group listener is available by pinging the floating IP address.

#### Manually move cluster resources

1. Log in to a cluster node using a sudo server over an SSH connection.
2. Verify that all cluster nodes are available.
3. Verify cluster status using the following command: `pcs resource --all`.

   Address all issues before attempting to start the cluster.
4. Run the following command: `pcs resource move <RESOURCE_NAME>-master <NODE_NAME>`
   
   Address all issues before attempting to start the cluster.

This command moves the resource agent to `<NODE_NAME>` and starts the resource. All cluster constraints will be applied. If the Microsoft SQL Server resource agent is moved, then the availability group listener will follow.

5. Verify cluster status using the following command: `pcs resource --all`.

The resource that was moved should be located on the `<NODE_NAME>`.

6. Clear temporary constraints using the following command: `pcs resource clear <RESOURCE_NAME>`.

---

**Stop cluster**

1. Log in to a cluster node using a sudo server over an SSH connection.
2. Verify that all cluster nodes are available.
3. Verify cluster status using the following command: `pcs resource --all`.

Address all issues before attempting to start the cluster.

4. Stop the cluster using the following command: `pcs cluster stop --ALL`. This will gracefully shut down all of the cluster nodes.

5. Verify the shut down status using the following command: `pcs status --all`.

This command should return that the cluster is no longer running.

---

**Recovery**

If a node is restarted from the operating system or the AWS Management Console, the Pacemaker node and its related services will not automatically start. This prevention protects the high availability database replicas from split-brain corruption.

The following steps are required to restore the cluster to normal operations.

1. Log in to a cluster node using a sudo server over an SSH connection.
2. Determine the node that was restarted using the following command: `pcs resource --ALL`. The restarted node will be offline.
3. Verify cluster status using the following command: `pcs resource --all`.

Address all issues before attempting to start the cluster.

4. Start the restarted node using the following command: `pcs cluster start --<NODE_NAME>`.

5. Verify cluster status using the following command: `pcs resource --all`.

Address all issues before attempting to start the cluster.

6. If the restarted node is the primary node of the cluster, then the Availability Group resource must be returned to the primary node.

7. Remove all temporary constraints using the following commands: `pcs resource clear <AG_RESOURCE>` and `pcs resource clear <AG_LISTENER>`.

8. Run the following command: `pcs resource move <RESOURCE_NAME> <PRI_NODE_NAME> --force`.

This command moves the resources to `<PRI_NODE_NAME>` and starts the resource. Any cluster constraints are applied. In this scenario, if the Microsoft SQL Server resource agent is moved, then the availability group listener follows.

9. Verify cluster status using the following command: `pcs resource --all`. The restarted node will be located on `<PRI_NODE_NAME>`.
Deploy an application with AWS Launch Wizard for SQL Server on RHEL

Topics
- Access AWS Launch Wizard (p. 185)
- Deploy AWS Launch Wizard on RHEL (p. 192)
- Post-deployment cluster tasks (p. 190)

Access AWS Launch Wizard

You can launch AWS Launch Wizard from the AWS Launch Wizard console.

Deploy AWS Launch Wizard on RHEL

The following steps guide you through a SQL Server application deployment with AWS Launch Wizard on the Red Hat Enterprise Linux (RHEL) platform after you have launched it from the console. For SQL Server deployments on RHEL, you must use an instance type built on the Nitro System. EBS volumes are exposed as NVMe block devices on instances built with the Nitro System. Device names that are specified for NVMe EBS volumes in a block device mapping are renamed using NVMe device names (/dev/nvme[0–26]n1). Launch Wizard deployments on RHEL do not support block devices on Xen-virtualized instances.

1. When you select Choose application from the AWS Launch Wizard landing page, you are directed to the Choose application wizard, where you are prompted to select the type of application that you want to deploy. Select Microsoft SQL Server, then Create deployment.
2. Under Review Permissions, Launch Wizard displays the AWS Identity and Access Management (IAM) role required for Launch Wizard to access other AWS services on your behalf. For more information about setting up IAM for Launch Wizard, see AWS Identity and Access Management (IAM) (p. 169). Choose Next.
3. On the Configure application settings page, select the Operating System on which you want to install SQL Server — in this case, Red Hat Enterprise Linux.
4. Deployment model. Choose High availability deployment to deploy your SQL Server Always On application across multiple Availability Zones.
5. You are prompted to enter specifications for the new deployment. The following tabs provide information about the input fields.

General

- Deployment name. Enter a unique application name for your deployment.
- Simple Notification Service (SNS) topic ARN (Optional). Specify an SNS topic where AWS Launch Wizard can send notifications and alerts. For more information, see the Amazon Simple Notification Service Developer Guide.
- Enable rollback on failed deployment. By default, if a deployment fails, your provisioned resources will not be rolled back/deleted. This default configuration helps you to troubleshoot errors at the resource level as you debug deployment issues. If you want your provisioned resources to be immediately deleted if a deployment fails, select the check box.

Connectivity

Enter your requirements for how you want to connect to your application instance and what kind of Virtual Private Cloud (VPC) you want to set up.
Key pair name

- Select an existing key pair from the dropdown list or create a new one. If you select **Create new key pair name** to create a new key pair, you are directed to the Amazon EC2 console. From there, under **Network and Security**, choose **Key Pairs**. Choose **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**.

**Important**

This is your only opportunity to save the private key file. Download it and save it in a safe place. You must provide the name of your key pair when you launch an instance, and provide the corresponding private key each time that you connect to the instance.

Return to the Launch Wizard console and choose the refresh button next to the **Key Pairs** dropdown list. The newly created key pair appears in the dropdown list. For more information about key pairs, see *Amazon EC2 Key Pairs and Windows Instances*.

Virtual Private Cloud (VPC). Choose whether you want to use an existing VPC or create a new VPC.

- **Select Virtual Private Cloud (VPC)** option. Choose the VPC that you want to use from the dropdown list. Your VPC must contain one public subnet. For HA deployments, it must also contain, at least, three private subnets. For single node deployments, it must contain one private subnet. The private subnets must have outbound connectivity to the internet and other AWS services (S3, CFN, SSM, Logs). We recommend that you enable this connectivity with a NAT Gateway. For more information about NAT Gateways, see **NAT Gateways** in the Amazon VPC User Guide.

- **Public Subnet**. Your VPC must contain one public subnet. For HA deployments it must also contain three private subnets. For single node deployments, it must contain one private subnet. Choose a public subnet for your VPC from the dropdown list. To continue, you must select the check box that indicates that the public subnet has been set up and each of the selected private subnets have outbound connectivity enabled.

To add a new public subnet

If the traffic of a subnet is routed to an internet gateway, the subnet is known as a public subnet. If, however, a subnet doesn’t have a route to the internet gateway, the subnet is known as a private subnet. To use an existing VPC that does not have a public subnet, you can add a new public subnet using the following steps.

- Follow the steps in **Creating a Subnet in the Amazon VPC User Guide** using the existing VPC you intend to use AWS Launch Wizard.

- To add an internet gateway to your VPC, follow the steps in **Attaching an Internet Gateway** in the Amazon VPC User Guide.

- To configure your subnets to route internet traffic through the internet gateway, follow the steps in **Creating a Custom Route Table** in the Amazon VPC User Guide. Use IPv4 format (0.0.0.0/0) for **Destination**.

- The public subnet should have the “auto-assign public IPv4 address” setting enabled. To enable this setting, follow the steps in **Modifying the Public IPv4 Addressing Attribute for Your Subnet** in the Amazon VPC User Guide.

- **Availability Zone (AZ) configuration**. You must choose at least three Availability Zones for High Availability (HA) deployments and one Availability Zone for single-node deployments, with one private subnet for each Availability Zone that you select. From the dropdown lists, select the **Availability Zones** within which you want to deploy your **primary**, **secondary**, and **configuration** nodes.
To create a private subnet

If a subnet doesn’t have a route to an internet gateway, the subnet is known as a private subnet. To create a private subnet, perform the following steps. We recommend that you enable the outbound connectivity for each of your selected private subnets using a NAT Gateway. To enable outbound connectivity from private subnets to public subnets, see the steps in Creating a NAT Gateway to create a NAT Gateway in your chosen public subnet. Then, follow the steps in Updating Your Route Table for each of your chosen private subnets.

- Follow the steps in Creating a Subnet in the Amazon VPC User Guide using the existing VPC you will use in AWS Launch Wizard.
- When you create a VPC, it includes a main route table by default. On the Route Tables page in the Amazon VPC console, you can view the main route table for a VPC by looking for Yes in the Main column. The main route table controls the routing for all subnets that are not explicitly associated with any other route table. If the main route table for your VPC has an outbound route to an internet gateway, then any subnet created using the previous step, by default, becomes a public subnet. To ensure the subnets are private, you may need to create one separate route table for all of your private subnets. This route table must not contain any routes to an internet gateway. Verify that all of the private subnets have the same route table association.
- Create new Virtual Private Cloud (VPC) option. Launch Wizard creates your VPC. You can optionally enter a VPC name tag.

SQL Server

SQL Server configuration

- User name and password. By default, Launch Wizard applies the user name sa. This system administrator account is used for SQL Server management. Create a complex password that is at least 8 characters long, and then reenter the password to verify it. See Password Policy for more information.
- Floating IP Address (HA and existing VPC deployments only). This field is available when you select a Virtual Private Cloud (VPC). The IP address that you enter is used as the endpoint for your SQL Server Availability Group listener. Launch Wizard creates a route from this IP address to the SQL primary node in your route table. Verify that the IP address is not already in use within your VPC and is outside of all of the provided subnet CIDRs.
- Amazon Machine Image (AMI). Select the version of Microsoft SQL Server Enterprise to deploy from the list of AMIs.
- SQL Server Edition. This field is available when you select a custom AMI. Choose the edition of SQL Server for the custom AMI: SQL Enterprise or SQL Standard.

Pacemaker cluster configuration (HA deployments only)

Pacemaker is a high-availability cluster resource manager. This software runs on a set of hosts, or cluster of nodes, to preserve integrity and minimize the downtime of selected services or resources. Pacemaker is maintained by the ClusterLabs community.

- Pacemaker cluster name. Enter a name to identify your pacemaker cluster.
- Pacemaker cluster username. By default, Launch Wizard applies the pacemaker username hacluster. This username is used to securely communicate between cluster nodes.
- Pacemaker cluster password. Create a complex password that is at least 8 characters long, and then reenter the password to verify it. See Password Policy for more information.
SQL - Pacemaker cluster connection settings (HA deployments only)

After you configure Pacemaker cluster and SQL Server, you must create a user in SQL Server to communicate with Pacemaker.

- SQL Pacemaker user name and password. Enter a user name for SQL Server to communicate with the Pacemaker cluster. Create a complex password that is at least 8 characters long, and then reenter the password to verify it. See Password Policy for more information.

- S3 location for node certificates. An Amazon S3 bucket location is required by the SQL nodes to share self-signed certificates with each other. Provide the bucket or object locations and verify that the names begin with launchwizard-.

Additional SQL Server settings (optional)

- Nodes. Enter a Primary SQL node name, a Secondary SQL node name, and a Configuration node name.
- Additional naming. Enter a Database name and an Availability group name.

6. When you are satisfied with your configuration selections, select Next. If you don't want to complete the configuration, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select Previous.

7. After configuring your application, you are prompted to define the infrastructure requirements for the new deployment on the Define infrastructure requirements page. The following tabs provide information about the input fields.

Storage and Compute

Infrastructure requirements based on infrastructure

You can choose to select your instances, storage and performance, and volume types, or to use AWS recommended resources. If you choose to use AWS recommended resources, you have the option of defining your high availability cluster needs. If no selections are made, default values are assigned.

- Number of instance cores. Choose the number of CPU cores for your infrastructure. The default value assigned is 4.
- Memory (GB). Choose the amount of RAM that you want to attach to your EC2 instances. The default value assigned is 4 GB.
- Type of storage drive. Select the storage drive type for the SQL data and tempdb volumes. The default value assigned is SSD.
- SQL Server throughput. Select the sustained SQL Server throughput that you need.
- Recommended resources. Launch Wizard displays the system-recommended resources based on your infrastructure selections. If you want to change the recommended resources, select different infrastructure requirements.

Infrastructure requirements based on instance type

You can choose to select your instance and volume type, or to use AWS recommended resources. If no selections are made, default values are assigned.

- Instance type. Select your preferred instance type from the dropdown list.
• **Volume type.** Choose your preferred EBS volume type. For more information about volume types, see Amazon EBS volume types

**Volume sizes**

• **Volume size.** Select the size of the SQL Server data volume in Gb for Temporary database, Data, and Backup volumes. SQL Server logs and data will be staged on the same data volume for this deployment. Make sure that you select an adequate size for the data volume.

**Tags-Optional**

You can provide optional custom tags for the resources Launch Wizard creates on your behalf. For example, you can set different tags for EC2 instances, EBS volumes, VPC, and subnets. If you select All, you can assign a common set of tags to your resources. Launch Wizard assigns tags with a fixed key LaunchWizardResourceGroupID and value that corresponds to the ID of the AWS resource group created for a deployment. Launch Wizard does not support custom tagging for root volumes.

**Estimated on-demand cost to deploy additional resources**

AWS Launch Wizard provides an estimate for application charges incurred to deploy the selected resources. The estimate updates each time you change a resource type in the wizard. The provided estimates are for general comparisons only. They are based upon On-Demand costs and your actual costs may be lower.

8. When you are satisfied with your infrastructure selections, select Next. If you don't want to complete the configuration, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the landing page. To go to the previous screen, select Previous.

9. On the Review and deploy page, review your configuration details. If you want to make changes, select Previous. To stop, select Cancel. When you select Cancel, all of the selections on the specification page are lost and you are returned to the service landing page. When you choose Deploy, you agree to the terms of the Note at the bottom of the page.

10. Launch Wizard validates the inputs and notifies you if you must update a specification.

11. When validation is complete, Launch Wizard deploys your AWS resources and configures your SQL Server Always On application. Launch Wizard provides you with status updates about the progress of the deployment on the Deployments page. From the Deployments page, you can view the list of current and previous deployments.

12. When your deployment is ready, a notification informs you that your SQL Server application is successfully deployed. If you have set up an SNS notification, you are also alerted through SNS. You can manage and access all of the resources related to your SQL Server Always On application by selecting the deployment, and then selecting Manage from the Actions dropdown list.

13. When the SQL Server Always On application is deployed, you can access your Amazon EC2 instances through the EC2 console. You can also use AWS SSM to manage your SQL Server Always On application for future updates and patches through built-in integration via resource groups.

**Post-deployment cluster tasks**

The Launch Wizard Pacemaker implementation includes three cluster nodes: primary, secondary, and configuration only. The primary node provides the Microsoft SQL Server for RHEL resource and the floating IP address. To ensure that the cluster operates correctly, some administrative tasks must be performed in a specific way. If these tasks are performed incorrectly, then Pacemaker may identify the activity as a resource failure and attempt to fail over the resources to the secondary node. If the resources are failed over to the secondary node, the cluster can remain in an unknown state, which can impact user access.
There are four primary tasks: **Start Cluster**, **Stop Cluster**, **Move Resources**, and **Recovery**. These tasks must be carried out by a sudo user with an SSH connection to any of the cluster nodes. Before performing any of these tasks, verify the cluster status using `pcs resource status --all`. This command returns all cluster issues. All issues must be addressed prior to performing any administrative tasks.

**Start cluster**

1. Log in to a cluster node using a sudo user over an SSH connection.
2. Verify that all cluster nodes are available.
3. Verify cluster status using the following command: `pcs resource --all`. Address all issues before attempting to start the cluster.
4. Start all cluster nodes using the following command: `pcs cluster start --all --wait`.
5. Verify that the cluster has started using the following command: `pcs resource --all`. The output provides information about the cluster nodes and cluster resources. All cluster nodes should be online and all resource agents should be visible and allocated to their assigned cluster nodes.
6. Verify that the availability group listener is available by pinging the floating IP address.

**Manually move cluster resources**

1. Log in to a cluster node using a sudo server over an SSH connection.
2. Verify that all cluster nodes are available.
3. Verify cluster status using the following command: `pcs resource --all`. Address all issues before attempting to start the cluster.
4. Run the following command: `pcs resource move <RESOURCE_NAME>-master <NODE_NAME> --force`. This command moves the resource agent to `<NODE_NAME>` and starts the resource. All cluster constraints will be applied. If the Microsoft SQL Server resource agent is moved, then the availability group listener will follow.
5. Verify cluster status using the following command: `pcs resource --all`. The resource that was moved should be located on the `<NODE_NAME>`.
6. Clear temporary constraints using the following command: `pcs resource clear <RESOURCE_NAME>`.

**Stop cluster**

1. Log in to a cluster node using a sudo server over an SSH connection.
2. Verify that all cluster nodes are available.
3. Verify cluster status using the following command: `pcs resource --all`. Address all issues before attempting to start the cluster.
4. Stop the cluster using the following command: `pcs cluster stop --ALL`. This will gracefully shut down all of the cluster nodes.
5. Verify the shut down status using the following command: `pcs status --all`. This command should return that the cluster is no longer running.
Recovery

If a node is restarted from the operating system or the AWS Management Console, the Pacemaker node and its related services will not automatically start. This prevention protects the high availability database replicas from split-brain corruption.

The following steps are required to restore the cluster to normal operations.

1. Log in to a cluster node using a sudo server over an SSH connection.
2. Determine the node that was restarted using the following command: `pcs resource --ALL`. The restarted node will be offline.
3. Verify cluster status using the following command: `pcs resource --all`. Address all issues before attempting to start the cluster.
4. Start the restarted node using the following command: `pcs cluster start --<NODE_NAME>`. Address all issues before attempting to start the cluster.
5. If the restarted node is the primary node of the cluster, then the Availability Group resource must be returned to the primary node.
6. Remove all temporary constraints using the following commands: `pcs resource clear <AG_RESOURCE>` and `pcs resource clear <AG_LISTENER>`.
7. Run the following command: `pcs resource move <RESOURCE_NAME> <PRI_NODE_NAME> --force`. This command moves the resources to `<PRI_NO_NAME>` and starts the resource. Any cluster constraints are applied. In this scenario, if the Microsoft SQL Server resource agent is moved, then the availability group listener follows.
8. Verify cluster status using the following command: `pcs resource --all`. The restarted node will be located on `<PRI_NO_NAME>`.

Manage application resources with AWS Launch Wizard for SQL Server

After your SQL Server Always On application is deployed, you can manage it by following these steps.

1. From the navigation pane, under Deployments, choose MS SQL Server.
2. From the Deployments — SQL page, select the deployment you want to manage and then select Actions. You can select to do the following:
   1. **Manage resources on the EC2 console.** You are taken to the Amazon EC2 console, where you can view and manage your SQL Server Always On application resources. For example, you can view and manage EC2, Amazon EBS, Active Directory, Amazon VPC, Subnets, NAT Gateways, and Elastic IPs. For SQL Server on Linux deployments, you can use AWS Systems Manager Session Manager to manage your deployed EC2 instances. For more information about SSM Session Manager, see AWS Systems Manager Session Manager.
   2. **Access SQL Server using RDGW instance (Windows deployments).** Connect to SQL Server via Remote Desktop Protocol. For more information, see Connecting to your Windows Instance in the User Guide for Windows Instances.
   3. **View resource group with SSM.** You are taken to the Systems Manager console to view your resource groups.
   4. **View SSM deployment template (Windows deployments).** You are taken to the Systems Manager console to view your documents.
5. **View CloudWatch application logs.** You are taken to CloudWatch Logs, where you can monitor, store, and access your SQL Server Always On application log files.

6. **View your CloudFormation template.** This is the CloudFormation template created by your most recent deployment, and it can be accessed through the CloudFormation console. For help with finding and using your CloudFormation template, see Viewing AWS CloudFormation Stack Data and Resources on the AWS Management Console.

7. If you have not set up monitoring for your application on CloudWatch Application Insights, you have the option to **Set up monitoring on CloudWatch Application Insights.** You are taken to the CloudWatch Application Insights console to set up monitoring for your application.

   If you have set up monitoring for your application on CloudWatch Application Insights, you can **View insights on Amazon CloudWatch.** You are taken to the application monitoring dashboards on the CloudWatch Application Insights console.

3. To delete a deployment, select the application that you want to delete and select **Delete.** You are prompted to confirm your action.

   **Important**
   You lose all specification settings for the SQL Server Always On application when you delete a deployment. Launch Wizard attempts to delete only the AWS resources that it created in your account as part of the deployment. If you created resources outside of Launch Wizard, for example, resources that reside in a VPC created by Launch Wizard, the deletion may fail. Launch Wizard does not delete any Active Directory objects in your Active Directory, nor any of the records in your DNS server. Launch Wizard has no control over your Active Directory domain user password over time, which is required to clean up Active Directory objects or DNS records. We recommend that you remove these entries from your Active Directory after Launch Wizard deletes the deployment. For key operations performed against your Active Directory resulting in new records or entries, see AWS Managed Active Directory (p. 167).

4. To drill down into details regarding your SQL Server Always On application resources, select the **Application name.** You can then view the **Deployment events** and **Configuration summary** details for your application by using the tabs at the top of the page.

---

**Manage Launch Wizard application resources with AWS Systems Manager Application Manager**

AWS Systems Manager Application Manager, a capability of AWS Systems Manager, helps you to investigate and remediate issues with your AWS resources that make up an application. Application Manager aggregates operations information from multiple AWS services and Systems Manager capabilities to a single console.

Application Manager automatically imports application resources created by Launch Wizard. From the Application Manager console, you can view operations details and perform operations tasks. You can also use runbooks, or SSM Automation documents, provided by Launch Wizard from the Application Manager console to manage or remediate issues with application components or resources.

For general information about AWS Systems Manager Application Manager, see AWS SSM Application Manager in the AWS Systems Manager User Guide.

The following information is specific to the management of Launch Wizard application resources from the Application Manager console.

**Topics**
- Use SSM Application Manager to run Automation workflows on your Launch Wizard applications (p. 200)
- Onboard existing applications (p. 201)
Use SSM Application Manager to run Automation workflows on your Launch Wizard applications

You can perform operations tasks and remediate issues with your Launch Wizard application resources by using AWS Systems Manager Automation runbooks.

Application Manager automatically imports all of your Launch Wizard resources and lists them in the Launch Wizard category. From the Application Manager console, choose Launch Wizard from the list of Applications. Select an application to view its information. On the Application information page, choose Start runbook. A dropdown list displays all of the runbooks available for your Launch Wizard application. This list includes runbooks provided by AWS, as well as any custom runbooks you own or are shared with you.

When you select a runbook, you are taken to the SSM Automation document console, where the resource group that makes up your application is preselected.

For descriptions of the runbooks provided by Launch Wizard, see AWS Launch Wizard Systems Manager Automation documents (p. 203).

Add custom runbooks

To add your own runbooks, you must modify the service setting value for the supported type.

1. The service setting value is a list of document Amazon Resource Names (ARNs). You can view this list using the following AWS Command Line Interface (AWS CLI) command, and adding the type to the setting id path.

   There are four supported types for which there are service settings:

   • AWS-SQLServerWindows
   • AWS-SQLServerLinux
   • AWS-SAP
   • AWS-SelfManagedActiveDirectory

   The following command lists the service settings for AWS-SQLServerWindows.

   ```
   aws ssm get-service-setting --setting-id /launchwizard/AWS-SQLServerWindows
   ```

   The following is the example output.

   ```json
   {
   "ServiceSetting": {
   "SettingId": "/launchwizard/AWS-SQLServerWindows",
   "LastModifiedDate": "2020-11-13T13:36:09.527000-05:00",
   "LastModifiedUser": "System",
   "Status": "Default"
   }
   }
   ```
2. You can modify the list of document ARNs by running the following command.

```bash
aws ssm update-service-setting \
   --setting-id /launchwizard/AWS-SQLServerWindows \
   --setting-value \
```

3. To reset the service setting value, run the following AWS CLI command. This command resets the service setting value for AWS-SQLServerWindows.

```bash
aws ssm reset-service-setting --setting-id /launchwizard/AWS-SQLServerWindows
```

The following is the example output.

```
{
   "ServiceSetting": {
      "SettingId": "/launchwizard/AWS-SQLServerWindows",
      "LastModifiedDate": "2020-11-13T13:36:09.527000-05:00",
      "LastModifiedUser": "System",
      "Status": "Default"
   }
}
```

The document lists correspond to the application type level. Therefore, when you add a new AWS-SQLServerWindows document, it will show up in all AWS-SQLServerWindows deployments. You can't add documents to a specific application.

**Note**
Verify that you use the correct Region for the added document ARNs.

### Onboard existing applications

When you deploy an application with Launch Wizard, the resource groups that make up the application are automatically assigned metadata showing that they are provisioned by Launch Wizard. Application Manager uses this metadata to display all of your resource groups and AWS CloudFormation stacks created by Launch Wizard on one page. When you deploy an application, Launch Wizard calls the `CreateOpsMetadata` API to assign the provisioning metadata.

**Onboard existing applications**

You can manually call the `CreateOpsMetadata` API using the AWS CLI so that existing application deployments appear on the Application Manager Launch Wizard page. The following example shows the `create-ops-metadata` AWS CLI command.

```bash
aws ssm create-ops-metadata \
   --resource-id "arn:aws:resource-groups:us-east-1:123456789012:group/LaunchWizard-SQLHAAlwaysOn-test" \
   --metadata '{"application-type": {"Value": "AWS-SQLServerWindows"}, "provisioned-by": {"Value": "AWS-LaunchWizard"}}'
```

You must provide the following information:
Onboard existing applications

- The resource group ARN of the resource that you want to be visible on the Launch Wizard page in Application Manager.
- A metadata JSON file that contains the application-type and provisioned-by key values. The application-type is the application type of the deployment, for example AWS-SQLServerWindows or AWS-SAP. The provisioned-by value is AWS-LaunchWizard.

When the command is successful, the output will be an OpsMetadataArn. If the output is an OpsMetadataAlreadyExistsException, then the resource group has already been tagged.

View all OpsMetadata values

You can call the ListOpsMetadata API to view all of your OpsMetadata values. To display only Launch Wizard-related metadata objects, you can use filtering. The following example shows the list-ops-metadata AWS CLI command.

```
aws ssm list-ops-metadata
  --filters '["Key":"provisioned-by","Values":["AWS-LaunchWizard"]]' \
  --max-results 20
```

The following is the example output.

```
{
  "OpsMetadataList": [
    {
      "ResourceId": "arn:aws:resource-groups:us-east-1:123456789012:group/LaunchWizard-SQLHAAlwaysOn-test",
      "LastModifiedDate": "2020-11-16T22:41:43.035000-05:00",
      "LastModifiedUser": "arn:aws:sts::123456789012:assumed-role/Admin",
      "CreationDate": "2020-11-16T22:41:43.035000-05:00"
    }
  ]
}
```

Filter by application type

The following example shows the list-ops-metadata AWS CLI command to filter by application type:

```
aws ssm list-ops-metadata
  --filters '["Key":"application-type","Values":["AWS-SQLServerWindows","AWS-SAP"]]' \
  --max-results 20
```

To get information about an OpsMetadataArn object, use the following command and enter the OpsMetadataArn.

```
aws ssm get-ops-metadata
```

The following is the example output.

```
{
  "ResourceId": "arn:aws:resource-groups:us-east-1:123456789012:group/LaunchWizard-SQLHAAlwaysOn-test",
  "Metadata": {
    "application-type": {
      "Value": "AWS-SQLServerWindows"
    }
  }
}
```
Patch management

You can automate the process of patching your Launch Wizard instances with security and other types of updates. From the Application information page of the Application Manager console, choose Patch. You are taken to the SSM Patch Manager console Patch now page, where patch management options for your application instances are preselected.

For more information about how Patch Manager determines which patches to install and how it installs them, see How Patch Manager operations work.

AWS Launch Wizard Systems Manager Automation documents

A Systems Manager Automation document defines the actions that Systems Manager performs on your managed instances and other AWS resources when an automation workflow runs. A document contains one or more steps that run in sequential order.

Launch Wizard provides predefined Automation documents that are maintained by AWS. This topic describes each of the predefined Automation documents provided for AWS Launch Wizard.

For more information about SSM Automation documents, see AWS SSM Automation in the AWS Systems Manager User Guide.

Launch Wizard-provided Automation documents:

- AWSSQLServer-DBCC (p. 203)
- AWSSQLServer-Backup (p. 204)
- AWSSQLServer-Index (p. 204)
- AWSSQLServer-Restore (p. 204)

AWSSQLServer-DBCC

The AWSSQLServer-DBCC Automation document includes the steps to perform database integrity checks on a specified database. You can control the type of database checks that are run. You can...
also adjust the execution parameters, such as specific tables to check, maximum CPU utilization, and more. For more information about the operations performed by DBCC checks, see the SQL Server documentation.

**AWSSQLServer-Backup**

The AWSSQLServer-Backup Automation document includes the steps to back up a specified database in either full, differential, or transactional mode. After the backup is completed, you can upload it to a specified folder within an S3 bucket.

The backup modes are defined as follows:

- **Full** — a complete backup of the database.
- **Differential** — the delta of changes since the last full backup.
- **Transactional** — a log of changes from the last full or differential backup, depending on the last backup type taken.

The following conditions must be met for the AWSSQLServer-Backup document to successfully back up a database:

- Backups must be staged to the local disk to run AWSSQLServer-Backup.
- The maximum size of the backup file for uploading to Amazon S3 is 500 GB or less.

**Required IAM actions that must be added to your IAM policy to successfully run AWSSQLServer-Backup:**

- `s3:UploadPart`
- `s3:CompleteMultipartUpload`
- `s3:CreateMultipartUpload`
- `s3:GetBucketPolicyStatus`
- `s3:GetBucketPolicyStatus`
- `s3:PutObject`

For more information about backup modes, see the Microsoft documentation.

**AWSSQLServer-Index**

The AWSSQLServer-Index Automation document includes steps to perform index maintenance operations on a specified database. You can choose a configuration, which includes the specific actions to take based on the level of fragmentation.

For more information about index maintenance operations, see the Microsoft documentation.

**AWSSQLServer-Restore**

The AWSSQLServer-Restore Automation document includes steps to download a backup database from a specified S3 bucket and folder to local storage. You can also optionally restore the backup to a copy of the database. The default behavior is to use the latest backup, and you can specify a time range to perform a point-in-time restore. The following conditions must be met for the AWSSQLServer-Restore document to successfully restore a database:

- The backup to use must have been performed by the AWSSQLServer-Backup document.
- There must be at least one full backup that occurred during the specified time range.
Required IAM actions that must be added to your IAM policy to successfully run
AWSQLServer-Restore:

- s3:UploadPart
- s3:CompleteMultipartUpload
- s3:CreateMultipartUpload
- s3:GetBucketPolicyStatus
- s3:GetBucketPolicyStatus
- s3:PutObject

Monitoring SQL Server Always On deployments

You can monitor your SQL Server Always On deployments using Amazon CloudWatch Application
Insights. When you select the option to monitor your deployment (p. 173) using the Launch Wizard
console, Application Insights identifies and sets up key metrics, logs, and alarms across your application
resources and technology stack for your Microsoft SQL Server database. Anomalies and errors are
detected and correlated as Application Insights continuously monitors metrics and logs. When errors
and anomalies are detected, Application Insights generates CloudWatch Events that you can use to set
up notifications or take action. To help with troubleshooting, Application Insights creates automated
dashboards for detected problems, which include correlated metric anomalies and log errors, along with
additional insights to point you to a possible root cause. Use the automated dashboards to take remedial
actions to keep your applications healthy and prevent end-user impact. You can also resolve problems
with AWS SSM OpsCenter using generated OpsItems.

For Microsoft SQL Server High Availability (HA) workloads, you can use CloudWatch Application
Insights to configure important counters, such as Mirrored Write Transaction/sec, Recovery Queue Length,
Transaction Delay, and Windows Event Logs on CloudWatch. You can also get automated insights
whenever a failover event or problem, such as restricted access to query a target database, is detected
with SQL HA workloads. See the Amazon CloudWatch Application Insights documentation for a complete
list of Logs and metrics supported by Application Insights.

High availability and security best practices for
AWS Launch Wizard for SQL Server

The application architecture created by AWS Launch Wizard supports AWS best practices for high
availability and security as promoted by the AWS Well-Architected Framework.

Topics
- High availability (p. 205)
- Automatic failover (p. 206)
- Security groups and firewalls (p. 206)

High availability

Using Amazon EC2, you can set the location of instances in multiple locations composed of AWS Regions
and Availability Zones. Regions are dispersed and located in separate geographic areas. Availability Zones
are distinct locations within a Region that are engineered to be isolated from failures in other Availability
Zones. Availability Zones provide inexpensive, low-latency network connectivity to other Availability
Zones in the same Region.
When you launch your instances in different Regions, you can set your SQL Server Always On application to be closer to specific customers, or to meet legal or other requirements. When you launch your instances in different Availability Zones, you can protect your SQL Server Always On applications from the failure of a single location. Windows Server Failover Clustering (WSFC) provides infrastructure features that complement the high availability and disaster recovery scenarios supported in the AWS Cloud.

**Automatic failover**

When you deploy AWS Launch Wizard with the default parameters, it configures a two-node, automatic failover cluster with a file share witness. An Always On Availability Group is deployed on this cluster with two availability replicas, as shown in the following diagram.

Launch Wizard implementation supports the following scenarios:

- Protection from the failure of a single instance
- Automatic failover between the cluster nodes
- Automatic failover between Availability Zones

The default implementation of Launch Wizard does not provide automatic failover in every case. For example, the failure of Availability Zone 1, which contains the primary node and file share witness, would prevent automatic failover to Availability Zone 2 because the cluster would fail as it loses quorum. In this scenario, you could follow manual disaster recovery steps that include restarting the cluster service and forcing quorum on the second cluster node (for example, WSFCNode2) to restore application availability. Launch Wizard also provides an option to deploy to three Availability Zones. This deployment option can mitigate the loss of quorum if a single node fails. However, you can select this option only in AWS Regions that include three or more Availability Zones. For a current list of supported Regions, see AWS Global Infrastructure.

**Security groups and firewalls**

Launch Wizard creates a number of security groups and rules for you. When Amazon EC2 instances are launched, they must be associated with a security group, which acts as a stateful firewall. You have complete control over the network traffic entering or leaving the security group. You can also build
granular rules that are scoped by protocol, port number, and source or destination IP address or subnet. By default, all outbound traffic from a security group is permitted. Inbound traffic, on the other hand, must be configured to allow the appropriate traffic to reach your instances.

The Securing the Microsoft Platform on Amazon Web Services whitepaper discusses the different methods for securing your AWS infrastructure. Recommendations include providing isolation between application tiers using security groups. We recommend that you tightly control inbound traffic in order to reduce the attack surface of your EC2 instances.

Domain controllers and member servers require several security group rules to allow traffic for services such as AD DS replication, user authentication, Windows Time services, and Distributed File System (DFS), among others. The WSFC nodes running SQL Server must permit several additional ports to communicate with each other. Finally, instances launched into the application server tier must establish SQL client connections to the WSFC nodes.

For a detailed list of port mappings, see the Security section of the Active Directory DS Quick Start guide.

In addition to security groups, the Windows Firewall must also be modified on the SQL Server instances. During the bootstrapping process, a script runs on each instance that opens the TCP ports 1433, 1434, 4022, 5022, 5023, and 135 on the Windows Firewall.

Troubleshoot AWS Launch Wizard for SQL Server

Each application in your account in the same AWS Region can be uniquely identified by the application name specified at the time of a deployment. The application name can be used to view the details related to the application launch.

For SQL Server deployments on Linux, you must use an instance type built on the Nitro System. EBS volumes are exposed as NVMe block devices on instances built with the Nitro System. Device names that are specified for NVMe EBS volumes in a block device mapping are renamed using NVMe device names (/dev/nvme[0–26]n1). Launch Wizard deployments on Linux do not support block devices on Xen-virtualized instances.

Contents
- Active Directory objects and DNS record clean up (deployment on Windows) (p. 207)
- Launch Wizard provisioning events (p. 208)
- CloudWatch Logs (p. 208)
- SSM Automation execution (p. 208)
- AWS CloudFormation stack (p. 209)
- Pacemaker on Ubuntu (deployment on Linux) (p. 209)
- SQL Server Management Studio (p. 210)
- Errors (p. 210)

Active Directory objects and DNS record clean up (deployment on Windows)

When you delete a deployment, you lose all specification settings for the SQL Server Always On application. Launch Wizard attempts to delete only the AWS resources that it created in your account as part of the deployment. If you created resources outside of Launch Wizard, for example, resources in a VPC created by Launch Wizard, the deletion can fail. Launch Wizard does not delete Active Directory objects in your Active Directory, nor does it delete any of the records in your DNS server. Launch Wizard has no control over your Active Directory domain user password over time, which is required to clean up
Active Directory objects or DNS records. We recommend that you remove these entries from your Active Directory after Launch Wizard deletes the deployment.

If the initial Active Directory objects or DNS records are not cleaned up, when you attempt to deploy Launch Wizard on an existing Active Directory using a deployment name that has already been used or availability group/listener/cluster name that has already been used, the deployment may fail with the following error.

**Error message**

System.Management.Automation.Remoting.PSRemotingTransportException: Connecting to remote server xxxxxxx failed with the following error message: WinRM cannot complete the operation. Verify that the specified computer name is valid, that the computer is accessible over the network, and that a firewall exception for the WinRM service is enabled and allows access from this computer. By default, the WinRM firewall exception for public profiles limits access to remote computers within the same local subnet.

To address this error, we recommend that you remove the initial entries from your Active Directory.

To clean up Active Directory Objects, run the following example PowerShell commands as a domain user with the appropriate authorization to perform these operations.

```powershell
$Pwd = ConvertTo-SecureString $password -AsPlainText –Force
$ADObject = Get-ADObject -Filter 'DNSHostName -eq "SQLnode.example.com"'
Remove-ADObject -Recursive -Identity $ADObject -Credential $cred
```

To remove a DNS Record, the name of the record that you want to delete (SQL Server node name), the DNS server FQDN, and the DNS zone within which the record is residing are required. The following are example PowerShell commands to perform the DNS record removal.

```powershell
$NodeDNS = Get-DnsServerResourceRecord -ZoneName $ZoneName -ComputerName $DNSServer -Node $NodeToDelete -RRType A -ErrorAction SilentlyContinue
Remove-DnsServerResourceRecord -ZoneName $ZoneName -ComputerName $DNSServer -InputObject $NodeDNS -Force
```

**Launch Wizard provisioning events**

Launch Wizard captures events from SSM Automation and AWS CloudFormation to track the status of an ongoing application deployment. If an application deployment fails, you can view the deployment events for this application by selecting **Deployments** from the navigation pane. A failed event shows a status of **Failed** along with a failure message.

**CloudWatch Logs**

Launch Wizard streams provisioning logs from all of the AWS log sources, such as AWS CloudFormation, SSM, and CloudWatch Logs. CloudWatch Logs for a given application name can be viewed on the CloudWatch console for the log group name LaunchWizard-APPLICATION_NAME and log stream ApplicationLaunchLog.

**SSM Automation execution**

Launch Wizard uses SSM Automation to provision SQL Server Always On applications. SSM Automation execution can be found in your account using the ssm describe-automation-executions API, and
adding document name prefix filters. Launch Wizard launches various automation documents in your account for validation and application provisioning. The following are the relevant filters for the `ssm describe-automation-executions` API.

- **Validation: Validate VPC connectivity**
  
  `LaunchWizard-Validate-VPC-Connectivity-APPLICATION_NAME-subnet_id`, where `subnet_id` is the subnet on which to perform the validation.

- **Validation: Validate credentials**
  
  `LaunchWizard-Validate-Credentials-APPLICATION_NAME`

- **Application Provisioning: Provisioning resources and Post Configuration**
  
  `LaunchWizard-SQLHAAlwaysOn-APPLICATION_NAME-Provision`

You can view the status of these SSM Automation executions. If any of them fail, you can view the cause of the failure.

**AWS CloudFormation stack**

Launch Wizard uses AWS CloudFormation to provision the infrastructure resources of an application. CloudFormation stacks can be found in your account using the CloudFormation `describe-stacks` API. Launch Wizard launches various stacks in your account for validation and application resource creation. The following are the relevant filters for the `describe-stacks` API.

- **Validation**
  
  `LaunchWizard-APPLICATION_NAME-checkCredentials-SSM_execution_id`

- **Validation**
  
  `LaunchWizard-APPLICATION_NAME-checkVPCConnectivity-SSM_execution_id`

- **Application resources**
  
  `LaunchWizard-APPLICATION_NAME`. This stack also has nested stacks for VPC, AD, the RDGW node, and SQL nodes.

You can view the status of these CloudFormation stacks. If any of them fail, you can view the cause of failure.

**Pacemaker on Ubuntu (deployment on Linux)**

To troubleshoot Pacemaker cluster resource issues, take the following actions as an administrator.

- Inspect the system log files for operating system errors and address the errors, as needed.
- Inspect the cluster log files for errors, including for errors that relate to Pacemaker, Corosync, or SQL Server. Check the log files carefully because the related services may provide only one or two related log entries.
- Verify resource configuration, and configuration of cluster-related functions.
  
  - The following commands display the configuration details:
    - To display all resources, use: `pcs resource show -full`.
    - Or, you can use: `pcs resource show <resource name>`.
    - The following command will display the cluster constraints: `pcs constraints -full`.
    - The following command displays the cluster properties: `pcs property list -all`.

209
• Manually start the resource with `debug-start`.
• Clear failed actions with the following command: `pcs resource cleanup <resource name>`.

### SQL Server Management Studio

If you encounter issues when you attempt to add databases with SQL Server Management Studio, perform the following to add databases to the availability group:

1. Log in to the primary node using SQL Server Management Studio (SSMS) and record the name of the availability group.
2. Verify that the database that you want to add to the availability group is backed up.
3. Add the database to the availability group by running the following command in SSMS:

   ```sql
   ALTER AVAILABILITY GROUP ag-name ADD DATABASE db
   ```

4. Refresh the availability group and verify that the database was created.

### Errors

#### Directory fails to create

- **Cause:** An internal service error has been encountered during the creation of the directory.
- **Solution:** Retry the operation. For this scenario, you must retry the deployment from the initial page of the Launch Wizard console.

#### Your requested instance type is not supported in your requested Availability Zone

- **Cause:** This failure might happen during the launch of either your RDGW instance or your SQL Server instance, or during the validation of the instances that Launch Wizard launches in your selected subnets.
- **Solution:** For this scenario, you must choose a different Availability Zone and retry the deployment from the initial page of the Launch Wizard console.

#### Validate connectivity for subnet. The following resource(s) failed to create: [ValidationNodeWaitCondition]

This failure can occur for multiple reasons. The following list shows known causes and solutions.

- **VPC or subnet configuration does not meet prerequisites**
  - **Cause:** This failure occurs when your VPC or subnet configuration does not meet the prerequisites documented in the VPC Connectivity Section under Deploy an application with AWS Launch Wizard for SQL Server on Windows (p. 173). If the failure message points to your selected public subnet, then the public subnet is not configured for outbound internet access. If the failure message points to one of your selected private subnets, then the specified private subnet does not have outbound connectivity.
  - **Solution:** Check that your VPC includes one public subnet and, at least, two private subnets. Your VPC must be associated with a DHCP Options Set to enable DNS translations to work. The private subnets must have outbound connectivity to the internet and other AWS services (S3, CFN, SSM, and Logs). We recommend that you enable this connectivity with a NAT Gateway. Note that, in the console, when you select a private subnet for the public subnet dropdown or you select a public subnet for the private subnet dropdown, you will encounter the same error. Please refer to the
VPC Connectivity section under Deploy an application with AWS Launch Wizard for SQL Server on Windows (p. 173) for more information about how to configure your VPC.

- **EC2 instance stabilization error**
  - **Cause:** Failure can occur if the EC2 instance used for validation fails to stabilize. When this happens, the EC2 instance is unable to communicate to the CloudFormation service to signal completions, resulting in *WaitCondition* errors.
  - **Solution:** Please contact AWS Support for assistance.
AWS Launch Wizard security

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to AWS Launch Wizard, see AWS Services in Scope by Compliance Program.

- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company’s requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS Launch Wizard. The following topics show you how to configure Launch Wizard to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Launch Wizard resources.

AWS Launch Wizard deploys Amazon EC2 instances into Amazon VPCs. For security information for Amazon EC2 and Amazon VPC, see the security sections in the Amazon EC2 Getting Started Guide and the Amazon VPC User Guide.

This section of the Launch Wizard User Guide provides security information that pertains to AWS Launch Wizard. For security topics specific to AWS Launch Wizard for SQL Server, see Security groups and firewalls (p. 206). For security topics specific to AWS Launch Wizard for SAP, see Security groups in AWS Launch Wizard for SAP (p. 148).

Launch Wizard security topics
- Infrastructure security in Launch Wizard (p. 212)
- Resilience in Launch Wizard (p. 212)
- Data protection in Launch Wizard (p. 213)
- Identity and Access Management for AWS Launch Wizard (p. 214)
- Update management in Launch Wizard (p. 215)
- AWS managed policies for AWS Launch Wizard (p. 215)

Infrastructure security in Launch Wizard

As a managed service, Launch Wizard is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

Resilience in Launch Wizard

The AWS global infrastructure is built around AWS Regions and Availability Zones. Regions provide multiple physically separated and isolated Availability Zones, which are connected through low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption.
Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

AWS Launch Wizard sets up an application across multiple Availability Zones to ensure automatic failover between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple datacenter infrastructures.

Data protection in Launch Wizard

The AWS shared responsibility model applies to data protection in AWS Launch Wizard. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the AWS Security Blog.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a Name field. This includes when you work with Launch Wizard or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Encryption with AWS managed keys and customer managed keys

AWS Launch Wizard for Active Directory, SQL Server, and SAP use the default AWS managed keys to encrypt Amazon EBS volumes. Launch Wizard for SAP also supports the use of customer managed keys that you have already created.

If you don't specify a customer managed key, Launch Wizard for SAP automatically creates an AWS managed key in your AWS account.

If you want to use a customer managed key for Launch Wizard for SAP, see the steps for adding permissions to your KMS key policy for Launch Wizard to use your KMS key at Add permissions to use AWS KMS keys (p. 84) in the Launch Wizard for SAP User Guide.
Creating your own customer managed CMK gives you more flexibility and control. For example, you can create, rotate, and disable customer managed keys. You can also define access controls and audit the customer managed keys that you use to protect your data. For more information about customer managed keys and AWS managed keys, see AWS KMS concepts in the AWS Key Management Service Developer Guide.

Identity and Access Management for AWS Launch Wizard

AWS Launch Wizard uses the following AWS managed policies to grant permissions to users and services.

- **AmazonEC2RolePolicyForLaunchWizard**

  AWS Launch Wizard creates an IAM role with the name AmazonEC2RoleForLaunchWizard in your account if the role already does not already exist in your account. If the role exists, the role is attached to the instance profile for the Amazon EC2 instances that Launch Wizard will launch into your account. This role is comprised of two IAM managed policies: AmazonSSMManagedInstanceCore and AmazonEC2RolePolicyForLaunchWizard.

  When you choose to deploy your SAP application with AWS Backint Agent for SAP HANA, you must attach the IAM inline policy provided in Step 2 of the AWS Identity and Access Management documentation for AWS Backint Agent for SAP HANA. This policy and instructions to attach the policy to the role are provided by Launch Wizard.

- **AmazonSSMManagedInstanceCore**

  This policy enables AWS Systems Manager service core functionality on Amazon EC2. For information, see Create an IAM Instance Profile for Systems Manager.

- **AmazonLaunchWizard_Fullaccess**

  This policy provides full access to AWS Launch Wizard and other required services.

- **AWSLambdaVPCAccessExecutionRole**

  This policy provides minimum permissions for a Lambda function to execute while accessing a resource within a VPC. These permissions include create, describe, delete network interfaces, and write permissions to CloudWatch Logs.

- **AmazonLambdaRolePolicyForLaunchWizardSAP**

  This policy provides minimum permissions to enable SAP provisioning scenarios on Launch Wizard. It allows invocation of Lambda functions to be able to perform certain actions, such as validation of route tables and perform pre-configuration and configuration tasks for HA mode enabling.

  - To run custom pre- and post-configuration deployment scripts, you must manually add the permissions provided in Add permissions to run custom pre- and post-deployment configuration scripts (p. 85) to the AmazonEC2RoleForLaunchWizard role.

  - To save generated artifacts from Launch Wizard for SAP to Amazon S3, and your S3 bucket name does not include the prefix launchwizard, you must attach the policy provided in Add permissions to save deployment artifacts to Amazon S3 (p. 86) to the IAM user.

  - To grant permissions for users to launch AWS Service Catalog products created with Launch Wizard for SAP, follow the steps in Set up to launch AWS Service Catalog products created with AWS Launch Wizard (p. 119).

  - To grant permissions to AWS Service Catalog to create a launch constraint for users who want to launch an AWS Service Catalog product created by Launch Wizard for SAP, follow the steps in Create a launch constraint (p. 119).
If you deploy domain controllers into an existing VPC with an existing Active Directory, Launch Wizard for Active Directory requires domain administrator credentials to be added to Secrets Manager in order to join your domain controllers to Active Directory and promote them. In addition, the following resource policy must be attached to the secret so that Launch Wizard can access the secret. Launch Wizard guides you through the process of attaching the required policy to your secret.

```
{
  "Version" : "2012-10-17",
  "Statement" : [{
    "Effect" : "Allow",
    "Principal" : {
      "AWS" :
      "arn:aws:iam::<account-id>:role/service-role/AmazonEC2RoleForLaunchWizard"
    },
    "Action" : [
      "secretsmanager:GetSecretValue",
      "secretsmanager:CreateSecret",
      "secretsmanager:GetRandomPassword"
    ],
    "Resource" : "*"
  }]
}
```

### Update management in Launch Wizard

We recommend that you regularly patch, update, and secure the operating system and applications on your EC2 instances. You can use AWS Systems Manager Patch Manager to automate the process of installing security-related updates for both the operating system and applications. Alternatively, you can use any automatic update services or recommended processes for installing updates that are provided by the application vendor.

### AWS managed policies for AWS Launch Wizard

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the IAM User Guide.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the ViewOnlyAccess AWS managed policy provides read-only access to many AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the IAM User Guide.

**Managed policies:**
- AWS managed policy: AmazonLaunchWizard_Fullaccess (p. 216)
• AWS managed policy: AmazonEC2RolePolicyForLaunchWizard (p. 226)
• AWS Launch Wizard updates to AWS managed policies (p. 230)

AWS managed policy: AmazonLaunchWizard_Fullaccess

You can attach the AmazonLaunchWizard_Fullaccess policy to your IAM identities.

This policy grants administrative permissions that allow full access to AWS Launch Wizard and other required services.

Permissions details

This policy includes the following permissions.

• launchwizard – Allows all Launch Wizard actions.
• applicationinsights – Allows all CloudWatch Application Insights actions. This permission is required so that an application can be tracked and configured by CloudWatch Application Insights, which provides Launch Wizard with more visibility and insight into the service through functionality such as monitoring and data analysis.
• route53 – Allows changing and listing resource record sets, listing hosted zones, and listing hosted zones by name. This is required so that scripts running on instances in a customer's account for SAP deployments can perform these actions.
• s3 – Allows all get or list operations for all resources, and allows for creation, deletion, and getting objects from a bucket, and putting objects in a bucket for certain Launch Wizard and SAP resources. This is required so that the Launch Wizard service can both view and update buckets and contents in Amazon S3 for tasks such as reading and storing scripts that are run on instances in its deployments.
• kms – Allows listing all AWS KMS keys and aliases. This is required so that Launch Wizard can view keys and aliases in a customer's account.
• cloudwatch – Allows all get, list, or describe actions for all resources, and allows Launch Wizard alarms and instance profiles to be created, updated, deleted, or described. This is required so that Launch Wizard can create and manage alarms to track metrics.
• ec2 – Allows creation of all security groups, authorization of ingress rules for all security groups, all get or describe operations, and creation of all VPCs, NAT/internet gateways, subnets, routes/route tables, and key pairs. Allows instances from the AWS CloudFormation stacks in Launch Wizard deployments to be stopped or terminated. Allows anything called from the Launch Wizard endpoint to perform other Amazon EC2 actions. This is required so that all EC2-related resources deployed from the Launch Wizard CloudFormation stacks can be appropriately created and managed.
• cloudformation – Allows all Launch Wizard and CloudWatch Application Insights CloudFormation stacks to be described and listed. Allows all get operations, all resources to be signaled, and all Launch Wizard stacks to be deleted. Allows all stacks to be created, and allows describe account limits, describe stack drift detection status, all list operations, and tagging of resources with all tag keys, starting with "Launch Wizard". This is required so that Launch Wizard can create CloudFormation stacks in a customer's account, so that the stacks are appropriately signaled, and so that a customer can view and delete those stacks.
• iam – Allows Launch Wizard EC2 roles and instance profiles to be created and deleted and attached/detached. Allows Launch Wizard EC2 and AWS Lambda roles and instance profiles to be passed a
role as long as it is passed to Lambda or EC2. Allows get operations for all roles or policies, all list operations, and all roles linked to Amazon EC2 Auto Scaling, CloudWatch Application Insights, or Amazon EventBridge to be created. This is required so that Launch Wizard can create necessary roles/users/policies and attach the appropriate roles/policies to them to ensure that resources in the Launch Wizard CloudFormation stacks and elsewhere in the service have the appropriate permissions.

- **autoscaling** – Allows Launch Wizard Auto Scaling groups and launch configurations to be created, deleted, and updated. This is required so that the Launch Wizard SQL CloudFormation stacks can perform these actions for the RDGW nodes in its deployments.

- **logs** – Allows all log groups or log streams to be created and deleted. Allows all write and read log events. This is required so that Launch Wizard can publish logs to a customer's account so that a customer can view the events from their deployments.

- **sns** – Allows Launch Wizard Amazon SNS topics to be created, deleted, subscribed to, and unsubscribed from. Allows all Amazon SNS subscriptions to be listed and messages to be published. This is required so that the Launch Wizard Amazon SNS queues to send signals between resources and Launch Wizard Lambda functions know when to proceed with steps in their event-based workflows.

- **resource-groups** – Allows resource groups whose names begin with "Launch Wizard" to be created, deleted, or listed. This is required so that Launch Wizard resources can be grouped together in a resource group, and so that the groups can be viewed or deleted.

- **ds** – Allows creation and deletion of a Microsoft Active Directory, adding IP routes, and all describe operations. This is required so that Active Directories can be created, deleted, and viewed in Launch Wizard SQL Server deployments, and so that IP routes can be added to them.

- **sqs** – Allows all queues with "SQS" in the name to be tagged, listed, created, and deleted. Allows any queue attributes to be set and read, and for the queue URL to be read and permissions added. This is required so that Launch Wizard SAP deployments can have a queue in the deployment on which these actions can be performed.

- **elasticfilesystem** – Allows all file systems to be created, deleted, and described, and for mount targets to be created, deleted, and described. This is required so that Launch Wizard SAP deployments can create Amazon EFS file systems in a customer's account with the appropriate mount targets.

- **lambda** – Allows AWS Lambda functions with "Launch Wizard" in the name to be created, deleted, read, and invoked. This is required so that Launch Wizard SAP deployments can perform some Lambda functions at the end of CloudFormation stacks for configuration in a customer's account or for parameter validation.

- **dynamodb** – Allows all tables with a name starting with "Launch Wizard" to be created, deleted, or described. This is required so that Launch Wizard scripts for SAP can publish events and metadata from the events of the running threads into a Amazon DynamoDB table in a customer's account.

- **secretsmanager** – Allows all secrets with a name starting with "Launch Wizard" to be created, deleted, and retrieved, all resources to be tagged or untagged, all resource policies to be created and deleted, and secret version IDs to be listed. Allows all random passwords to be generated and all secrets to be listed. This is required so that secrets can be created in a customer's account to perform operations, such as decrypting a password in order to RDP into an instance from their deployment.

- **fsx** – Allows Amazon FSx file systems to be created by all resources. Allows describing file system properties, listing all tags on the Amazon FSx file share, adding and removing tags, and deleting file systems where tags include LaunchWizard in the name.

- **servicecatalog** – Allows for the creation of an AWS Service Catalog portfolio and product. Allows for the creation of a LaunchConstraint. Allows for the association between product and portfolio, as well as for the association between the IAM principal of a user and a portfolio.

- **ssm** – Allows for all get, list, tag, execute, and delete operations for all SSM resources. This is required so that Launch Wizard can create, run, and delete SSM resources on behalf of the customer to configure their Amazon EC2 instances for application provisioning.

```json
{
    "Version": "2012-10-17",
}
```
"Statement": [
    {  
        "Effect": "Allow",
        "Action": "applicationinsights:*",
        "Resource": "*"
    },
    {  
        "Effect": "Allow",
        "Action": "resource-groups:List*",
        "Resource": "*"
    },
    {  
        "Effect": "Allow",
        "Action": [  
            "route53:ChangeResourceRecordSets",
            "route53:GetChange",
            "route53:ListResourceRecordSets",
            "route53:ListHostedZones",
            "route53:ListHostedZonesByName"
        ],
        "Resource": "*"
    },
    {  
        "Effect": "Allow",
        "Action": [  
            "s3:ListAllMyBuckets",
            "s3:ListBucket",
            "s3:GetBucketLocation"
        ],
        "Resource": "*"
    },
    {  
        "Effect": "Allow",
        "Action": [  
            "kms:ListKeys",
            "kms:ListAliases"
        ],
        "Resource": "*"
    },
    {  
        "Effect": "Allow",
        "Action": [  
            "cloudwatch:List*",
            "cloudwatch:Get*",
            "cloudwatch:Describe*"
        ],
        "Resource": "*"
    },
    {  
        "Effect": "Allow",
        "Action": [  
            "ec2:CreateInternetGateway",
            "ec2:CreateNatGateway",
            "ec2:CreateVpc",
            "ec2:CreateKeyPair",
            "ec2:CreateRoute",
            "ec2:CreateRouteTable",
            "ec2:CreateSubnet"
        ],
        "Resource": "*"
    }]
}
"ec2:AssignPrivateIpAddresses",
"ec2:AssociateAddress",
"ec2:CreateDhcpOptions",
"ec2:CreateEgressOnlyInternetGateway",
"ec2:CreateNetworkInterface",
"ec2:CreateVolume",
"ec2:CreateVpcEndpoint",
"ec2:CreateTags",
"ec2:DeleteTags",
"ec2:RunInstances",
"ec2:StartInstances",
"ec2:ModifyInstanceAttribute",
"ec2:ModifySubnetAttribute",
"ec2:ModifyVolumeAttribute",
"ec2:ModifyVpcAttribute",
"ec2:AssociateDhcpOptions",
"ec2:AssociateSubnetCidrBlock",
"ec2:AttachInternetGateway",
"ec2:AttachNetworkInterface",
"ec2:AttachVolume",
"ec2:DeleteDhcpOptions",
"ec2:DeleteInternetGateway",
"ec2:DeleteKeyPair",
"ec2:DeleteNatGateway",
"ec2:DeleteSecurityGroup",
"ec2:DeleteVolume",
"ec2:DeleteVpc",
"ec2:DetachInternetGateway",
"ec2:DetachVolume",
"ec2:DeleteSnapshot",
"ec2:AssociateRouteTable",
"ec2:AssociateVpcCidrBlock",
"ec2:DeleteNetworkAcl",
"ec2:DeleteNetworkInterface",
"ec2:DeleteNetworkInterfacePermission",
"ec2:DeleteRoute",
"ec2:DeleteRouteTable",
"ec2:DeleteSubnet",
"ec2:DetachNetworkInterface",
"ec2:DisassociateAddress",
"ec2:DisassociateVpcCidrBlock",
"ec2:GetLaunchTemplateData",
"ec2:ModifyNetworkInterfaceAttribute",
"ec2:ModifyVolume",
"ec2:AuthorizeSecurityGroupEgress",
"ec2:GetConsoleOutput",
"ec2:GetPasswordData",
"ec2:ReleaseAddress",
"ec2:ReplaceRoute",
"ec2:ReplaceRouteTableAssociation",
"ec2:RevokeSecurityGroupEgress",
"ec2:RevokeSecurityGroupIngress",
"ec2:DisassociateIamInstanceProfile",
"ec2:DisassociateRouteTable",
"ec2:DisassociateSubnetCidrBlock",
"ec2:ModifyInstancePlacement",
"ec2:DeletePlacementGroup",
"ec2:CreatePlacementGroup",
"elasticfilesystem:DeleteFileSystem",
"elasticfilesystem:DeleteMountTarget",
"ds:AddIpRoutes",
"ds:CreateComputer",
"ds:CreateMicrosoftAD",
"ds:DeleteDirectory",
"servicecatalog:AssociateProductWithPortfolio",
"cloudformation:GetTemplateSummary"
"sts:GetCallerIdentity"
],
"Resource": "**",
"Condition": {
  "ForAnyValue:StringEquals": {
    "aws:CalledVia": "launchwizard.amazonaws.com"
  }
}
},
{
  "Effect": "Allow",
  "Action": [
    "cloudformation:DescribeStack*",
    "cloudformation:Get*",
    "cloudformation:ListStacks",
    "cloudformation:SignalResource",
    "cloudformation:DeleteStack"
  ],
  "Resource": [
    "arn:aws:cloudformation:*:*:stack/LaunchWizard*/*",
    "arn:aws:cloudformation:*:*:stack/ApplicationInsights*/*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:StopInstances",
    "ec2:TerminateInstances"
  ],
  "Resource": "**",
  "Condition": {
    "StringLike": {
      "ec2:ResourceTag/aws:cloudformation:stack-id": "arn:aws:cloudformation:*:*:stack/LaunchWizard-*/*"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateInstanceProfile",
    "iam:DeleteInstanceProfile",
    "iam:RemoveRoleFromInstanceProfile",
    "iam:AddRoleToInstanceProfile"
  ],
  "Resource": [
    "arn:aws:iam::*:role/service-role/AmazonEC2RoleForLaunchWizard*",
    "arn:aws:iam::*:instance-profile/LaunchWizard*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iam:PassRole"
  ],
  "Resource": [
    "arn:aws:iam::*:role/service-role/AmazonEC2RoleForLaunchWizard*",
    "arn:aws:iam::*:instance-profile/LaunchWizard*"
  ],
  "Condition": {
    "StringEqualsIfExists": {
      "iam:PassedToService": [
        "lambda.amazonaws.com",
        "ec2.amazonaws.com"
      ]
    }
  }
}
"Effect": "Allow",
"Action": [
  "autoscaling:AttachInstances",
  "autoscaling:CreateAutoScalingGroup",
  "autoscaling:CreateLaunchConfiguration",
  "autoscaling:DeleteAutoScalingGroup",
  "autoscaling:DeleteLaunchConfiguration",
  "autoscaling:UpdateAutoScalingGroup",
  "logs:CreateLogStream",
  "logs:DeleteLogGroup",
  "logs:DeleteLogStream",
  "logs:DescribeLog*",
  "logs:PutLogEvents",
  "resource-groups:CreateGroup",
  "resource-groups:DeleteGroup",
  "sns:ListSubscriptionsByTopic",
  "sns:Publish",
  "ssm:DeleteDocument",
  "ssm:DeleteParameter*",
  "ssm:DescribeDocument**",
  "ssm:GetDocument",
  "ssm:GetParameter"
],
"Resource": [
  "arn:aws:resource-groups:*:*:group/LaunchWizard*",
  "arn:aws:ssm:*:*:parameter/LaunchWizard*",
  "arn:aws:ssm:*:*:document/LaunchWizard*",
  "arn:aws:logs:*:*:log-group:*:*:*",
  "arn:aws:logs:*:*:log-group:LaunchWizard*"
]}
},
{  "Effect": "Allow",
  "Action": [
    "ssm:GetDocument",
    "ssm:SendCommand"
  ],
  "Resource": [
    "arn:aws:ssm:*:*:document/AWS-RunShellScript"
  ]
},

{  "Effect": "Allow",
  "Action": [
    "ssm:SendCommand"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:instance/*
  ],
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/aws:cloudformation:stack-id": "arn:aws:cloudformation:*:*:stack/LaunchWizard-*/**"
    }
  }
},

{  "Effect": "Allow",
  "Action": [
    "ssm:SendCommand"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:instance/*
  ],
  "Condition": {
    "StringLike": {
      "aws:ResourceTag/aws:cloudformation:stack-id": "arn:aws:cloudformation:*:*:stack/LaunchWizard-*/**"
    }
  }
}
"Action": [
  "logs:DeleteLogStream",
  "logs:GetLogEvents",
  "logs:PutLogEvents",
  "ssm:AddTagsToResource",
  "ssm:DescribeDocument",
  "ssm:GetDocument",
  "ssm:ListTagsForResource",
  "ssm:RemoveTagsFromResource"
],
"Resource": [
  "arn:aws:logs:*:*:log-group:*:*:*",
  "arn:aws:logs:*:*:log-group:LaunchWizard*",
  "arn:aws:ssm:*:*:parameter/LaunchWizard*",
  "arn:aws:ssm:*:*:document/LaunchWizard*"
],
"Effect": "Allow",
"Action": [
  "autoscaling:Describe*",
  "cloudformation:DescribeAccountLimits",
  "cloudformation:DescribeStackDriftDetectionStatus",
  "cloudformation:List*",
  "cloudformation:ValidateTemplate",
  "ds:Describe*",
  "ds:ListAuthorizedApplications",
  "ec2:Describe*",
  "ec2:Get*",
  "iam:GetRole",
  "iam:GetRolePolicy",
  "iam:GetUser",
  "iam:GetPolicyVersion",
  "iam:GetPolicy",
  "iam:List*",
  "logs:CreateLogGroup",
  "logs:GetLogDelivery",
  "logs:GetLogRecord",
  "logs:ListLogDeliveries",
  "resource-groups:Get*",
  "resource-groups:List*",
  "servicequotas:GetServiceQuota",
  "servicequotas:ListServiceQuotas",
  "sns:ListSubscriptions",
  "sns:ListTopics",
  "ssm:CreateDocument",
  "ssm:DescribeAutomation*",
  "ssm:DescribeInstanceInformation",
  "ssm:DescribeParameters",
  "ssm:GetAutomationExecution",
  "ssm:GetCommandInvocation",
  "ssm:GetParameter*",
  "ssm:GetConnectionStatus",
  "ssm:ListCommand*",
  "ssm:ListDocument*",
  "ssm:ListInstanceAssociations",
  "ssm:SendAutomationSignal",
  "tag:Get*"
],
"Resource": "*"]
"cloudwatch:DescribeAlarms"
],
"Resource": [
  "arn:aws:cloudwatch:*:alarm:LaunchWizard*",
  "arn:aws:iam::*:instance-profile/LaunchWizard*"
],
"Effect": "Allow",
"Action": [
  "cloudformation:CreateStack",
  "route53:ListHostedZones",
  "ec2:CreateSecurityGroup",
  "ec2:AuthorizeSecurityGroupIngress",
  "elasticfilesystem:DescribeFileSystems",
  "elasticfilesystem:CreateFileSystem",
  "elasticfilesystem:CreateMountTarget",
  "elasticfilesystem:DescribeMountTargets",
  "elasticfilesystem:DescribeMountTargetSecurityGroups"
],
"Resource": "*"
},
{
"Effect": "Allow",
"Action": [
  "s3:GetObject",
  "s3:PutObject"
],
"Resource": [
  "arn:aws:s3:::launchwizard*",
  "arn:aws:s3:::aws-sap-data-provider/config.properties"
],
"Effect": "Allow",
"Action": "cloudformation:TagResource",
"Resource": "*",
"Condition": {
  "ForAllValues:StringLike": {
    "aws:TagKeys": "LaunchWizard*"
  }
}
},
{
"Effect": "Allow",
"Action": [
  "s3:CreateBucket",
  "s3:PutBucketVersioning",
  "s3:DeleteBucket",
  "lambda:CreateFunction",
  "lambda:DeleteFunction",
  "lambda:GetFunction",
  "lambda:GetFunctionConfiguration",
  "lambda:InvokeFunction"
],
"Resource": [
  "arn:aws:lambda::*:function:LaunchWizard*",
  "arn:aws:s3:::launchwizard*"
],
"Effect": "Allow",
"Action": [
  "dynamodb:CreateTable",
  "dynamodb:DescribeTable",
  "dynamodb:UpdateTable"
"dynamodb:DeleteTable",

"Resource": "arn:aws:dynamodb:::table/LaunchWizard",

},

{ "Effect": "Allow",
"Action": [ "secretsmanager:CreateSecret",
"secretsmanager:DeleteSecret",
"secretsmanager:TagResource",
"secretsmanager:UntagResource",
"secretsmanager:PutResourcePolicy",
"secretsmanager:DeleteResourcePolicy",
"secretsmanager:ListSecretVersionIds",
"secretsmanager:GetSecretValue"
],
"Resource": "arn:aws:secretsmanager:::secret:LaunchWizard" },

{ "Effect": "Allow",
"Action": [ "secretsmanager:GetRandomPassword",
"secretsmanager:ListSecrets"
],
"Resource": "*
",

{ "Effect": "Allow",
"Action": [ "ssm:CreateOpsMetadata"
],
"Resource": "*
",

{ "Effect": "Allow",
"Action": "ssm:DeleteOpsMetadata",
"Resource": "arn:aws:ssm:::opsmetadata/aws/ssm/LaunchWizard"
 },

{ "Effect": "Allow",
"Action": [ "sns:CreateTopic",
"sns:DeleteTopic",
"sns:Subscribe",
"sns:Unsubscribe"
],
"Resource": "arn:aws:ssm:::LaunchWizard"
 },

{ "Effect": "Allow",
"Action": [ "fsx:UntagResource",
"fsx:TagResource",
"fsx:DeleteFileSystem",
"fsx:ListTagsForResource"
],
"Resource": "*
",

"Condition": {
"StringLike": {
"aws:ResourceTag/Name": "LaunchWizard"
}
}
},

{ "Effect": "Allow",
"Action": [ 

}
"fsx:CreateFileSystem"
],
"Resource": "*",
"Condition": {
  "StringLike": {
    "aws:RequestTag/Name": [
      "LaunchWizard*"
    ]
  }
},

{  
  "Effect": "Allow",
  "Action": [  
    "fsx:DescribeFileSystems"
  ],
  "Resource": "*"
},

{  
  "Effect": "Allow",
  "Action": [  
    "servicecatalog:CreatePortfolio",
    "servicecatalog:DescribePortfolio",
    "servicecatalog:CreateConstraint",
    "servicecatalog:CreateProduct",
    "servicecatalog:AssociatePrincipalWithPortfolio",
    "servicecatalog:CreateProvisioningArtifact"
  ],
  "Resource": [  
    "arn:aws:servicecatalog:*:*:*/*",
    "arn:aws:catalog:*:*:*/*"
  ],
  "Condition": {
    "ForAnyValue:StringEquals": {  
      "aws:CalledVia": "launchwizard.amazonaws.com"
    }
  }
}
}

**Note**

arn:aws:s3:::launchwizard* and arn:aws:s3:::launchwizard*/* are redundant permissions. Both permissions are present for historical purposes and do not impact security.

**AWS managed policy:**

**AmazonEC2RolePolicyForLaunchWizard**

You can attach the `AmazonEC2RolePolicyForLaunchWizard` policy to your IAM identities.

This policy grants administrative permissions that allow all AWS Launch Wizard actions to be performed.

**Permissions details**

This policy includes the following permissions.

- `launchwizard` – Allows all Launch Wizard actions.
• **ec2** – Allows starting, stopping, and rebooting instances, and attaching volumes to all instances with the `LaunchWizardResourceGroupID` tag. Allows replacing route table for all instances with the `LaunchWizardApplicationType` resource tag. Allows all resources to describe and associate IP addresses, describe instances, images, Regions, volumes, and route tables, and modify instance attributes for all resources. Allows creating tags and volumes for all resources with the `LaunchWizardResourceType` or `LaunchWizardResourceGroupID` tags.

• **cloudwatch** – Allows for getting and writing metrics to CloudWatch. This is required so that CloudWatch can write logs for all resources.

• **s3** – Allows all get or list operations for all resources, and allows for creation, deletion, and getting objects from a bucket, and putting objects in a bucket for certain Launch Wizard and SAP resources. This is required so that the Launch Wizard service can both view and update buckets and contents in Amazon S3 for tasks such as reading and storing scripts that are run on instances in its deployments.

• **ssm** – Allows send commands to all Amazon EC2 instances with the `LaunchWizardApplicationType` resource tag. Allows getting a document. These actions are required to run the Backint install agent SSM document for SAP.

• **logs** – Allows all log groups or log streams for all write and read log events. This is required so that Launch Wizard can publish logs to a customer’s account so that a customer can view the events from their deployments.

• **cloudformation** – Allows all Launch Wizard and CloudWatch Application Insights CloudFormation stacks to be described and listed. Allows all get operations and for all resources to be signaled. This is required so that the stacks are appropriately signaled by CloudFormation.

• **dynamodb** – Allows all tables with a name starting with "Launch Wizard" to be created, deleted, or described. This is required so that Launch Wizard scripts for SAP can publish events and metadata from the events of the running threads into a Amazon DynamoDB table in a customer’s account.

• **sqs** – Allows sending and receiving messages from Amazon SQS queues. This is required so that Launch Wizard SAP deployments can have a queue in the deployment on which these actions can be performed.

• **iam** – Allows Launch Wizard EC2 roles and instance profiles to be created and deleted and attached/detached. Allows Launch Wizard EC2 and AWS Lambda roles and instance profiles to be passed a role as long as it is passed to Lambda or EC2. Allows get operations for all roles or policies, all list operations, and all roles linked to Amazon EC2 Auto Scaling, CloudWatch Application Insights, or Amazon EventBridge to be created. This is required so that Launch Wizard can create necessary roles/users/policies and attach the appropriate roles/policies to them to ensure that resources in the Launch Wizard CloudFormation stacks and elsewhere in the service have the appropriate permissions.

• **fsx** – Allows describing file systems and listing tags on file systems on any Amazon FSx resource tagged with the `LaunchWizard` tag. This is required so that Launch Wizard can retrieve the FSX DNS and administration endpoints to create the FCI SQL cluster.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:RebootInstances",
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:volume/*",
        "arn:aws:ec2:*:*:instance/*"
      ],
      "Condition": {
        "StringLike": {
          "ec2:*:instance/*
        }
      }
    }
  ]
}
```
"ec2:ResourceTag/LaunchWizardResourceGroupID": "*"

},

"Effect": "Allow",
"Action": [
  "ec2:ReplaceRoute"
],
"Resource": "arn:aws:ec2:*:*:route-table/**",
"Condition": {
  "StringLike": {
    "ec2:ResourceTag/LaunchWizardApplicationType": "*"
  }
}

},

"Effect": "Allow",
"Action": [
  "ec2:DescribeAddresses",
  "ec2:AssociateAddress",
  "ec2:DescribeInstances",
  "ec2:DescribeImages",
  "ec2:DescribeRegions",
  "ec2:DescribeVolumes",
  "ec2:DescribeRouteTables",
  "ec2:ModifyInstanceState",
  "cloudwatch:GetMetricStatistics",
  "cloudwatch:PutMetricData",
  "ssm:GetCommandInvocat"
],
"Resource": "*"

},

"Effect": "Allow",
"Action": [
  "ec2:CreateTags",
  "ec2:CreateVolume"
],
"Resource": "arn:aws:ec2:*:*:volume/**",
"Condition": {
  "ForAllValues:StringEquals": {
    "aws:TagKeys": [
      "LaunchWizardResourceGroupID",
      "LaunchWizardApplicationType"
    ]
  }
}

},

"Effect": "Allow",
"Action": [
  "s3:GetObject",
  "s3:ListBucket",
  "s3:PutObject",
  "s3:PutObjectTagging",
  "s3:GetBucketLocation",
  "logs:PutLogEvents",
  "logs:DescribeLogGroups",
  "logs:DescribeLogStreams"
],
"Resource": [
  "arn:aws:s3:::*:*",
  "arn:aws:s3:::launchwizard**",
  "arn:aws:s3:::aws-sap-data-provider/config.properties"
]


```json
{
  "Effect": "Allow",
  "Action": [ "logs:Create***",
               "ec2:Describe***",
               "cloudformation:DescribeStackResources",
               "cloudformation:SignalResource",
               "cloudformation:DescribeStackResource",
               "cloudformation:DescribeStacks"
             ],
  "Resource": "arn:aws:logs:*:*:*",
  "Condition": {
    "StringEquals": [ "aws:TagKeys": "LaunchWizardResourceGroupID"
                      ]
  }
},
{
  "Effect": "Allow",
  "Action": [ "ec2:Describe***",
               "cloudformation:DescribeStackResources",
               "cloudformation:SignalResource",
               "cloudformation:DescribeStackResource",
               "cloudformation:DescribeStacks"
             ],
  "Resource": "***",
  "Condition": {
    "ForAllValues:StringLength": [ "aws:TagKeys": "LaunchWizardResourceGroupID"
                                   ]
  }
},
{
  "Effect": "Allow",
  "Action": [ "dynamodb:BatchGetItem",
               "dynamodb:PutItem",
               "sqs:ReceiveMessage",
               "sqs:SendMessage",
               "dynamodb:Scan",
               "s3:ListBucket",
               "dynamodb:Query",
               "dynamodb:UpdateItem",
               "dynamodb:DeleteTable",
               "dynamodb:CreateTable",
               "s3:GetObject",
               "dynamodb:DescribeTable",
               "s3:GetBucketLocation",
               "dynamodb:UpdateTable"
             ],
  "Resource": [ "arn:aws:s3:::launchwizard***",
                 "arn:aws:dynamodb::*::*:table/LaunchWizard***",
                 "arn:aws:sqs::*::*:LaunchWizard***"
               ]
},
{
  "Effect": "Allow",
  "Action": [ "ssm:SendCommand",
               "ssm:GetDocument"
             ],
  "Resource": [ "arn:aws:ssm::*:*:document/AWSSAP-InstallBackint"
             ]
}
```
AWS Launch Wizard updates to AWS managed policies

View details about updates to AWS managed policies for AWS Launch Wizard since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the AWS Launch Wizard Document history page.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmazonLaunchWizard_Fullaccess (p. 216)</td>
<td>– Update to an existing policy</td>
<td>April 12, 2022</td>
</tr>
<tr>
<td></td>
<td>AWS Launch Wizard restricted ssm actions to only documents containing the LaunchWizard prefix and called by the Launch Wizard service to improve the security of this managed policy.</td>
<td></td>
</tr>
<tr>
<td>AmazonLaunchWizard_Fullaccess (p. 216)</td>
<td>– Update to an existing policy</td>
<td>February 9, 2022</td>
</tr>
<tr>
<td></td>
<td>AWS Launch Wizard restricted ssm:sendCommand actions to only the arn:aws:ec2:***:instance/ * resource and to resources with the tag keys aws:cloudformation:stack-id to improve the security of this managed policy.</td>
<td></td>
</tr>
<tr>
<td>AmazonLambdaRoleForLaunchWizard</td>
<td>– Policy deprecation</td>
<td>February 7, 2022</td>
</tr>
<tr>
<td></td>
<td>AWS Launch Wizard deprecated the AmazonLambdaRoleForLaunchWizard policy because it is no longer used by the service.</td>
<td></td>
</tr>
<tr>
<td>AmazonEC2RolePolicyForLaunchWizard</td>
<td>– Update to an existing policy</td>
<td>February 7, 2022</td>
</tr>
<tr>
<td></td>
<td>AWS Launch Wizard restricted the ec2:CreateTags and ec2:CreateVolume actions to the arn:aws:ec2:***:volume/</td>
<td></td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>* resource to prohibit tagging of other resources with the tag keys LaunchWizardResourceGroupID and LaunchWizardApplicationType to improve the security of the managed policy.</td>
<td>August 30, 2021</td>
<td></td>
</tr>
<tr>
<td><strong>AmazonLaunchWizard_Fullaccess</strong> (p. 216) – Update to an existing policy</td>
<td>AWS Launch Wizard added new policies to support the creation of AWS Service Catalog portfolios and products with Launch Wizard. AWS Launch Wizard will perform servicecatalog:CreatePortfolio, servicecatalog:DescribePortfolio, servicecatalog:CreateConstraint, servicecatalog:CreateProduct, servicecatalog:AssociatePrincipalWithPortfolio, servicecatalog:CreateProvisioningArtifact, and servicecatalog:AssociateProductWithPortfolio actions on AWS Service Catalog resources when they are called by Launch Wizard.</td>
<td>August 30, 2021</td>
</tr>
<tr>
<td><strong>AmazonEC2RolePolicyForLaunchWizard</strong> (p. 226) – Update to an existing policy</td>
<td>AWS Launch Wizard added new policies to support FSx creation with Launch Wizard. AWS Launch Wizard will perform fsx:DescribeFileSystems and fsx:ListTagsForResource actions on all resources created by Launch Wizard with LaunchWizard* in the tag when they are called via launchwizard.amazonaws.com to enable this support.</td>
<td>May 21 2021</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>AmazonLaunchWizard_Fullaccess</td>
<td>AWS Launch Wizard added new permissions to allow PlacementGroup for SAP HANA scale-out scenarios. AWS Launch Wizard will perform ec2:ModifyInstancePlacement, ec2:DeletePlacementGroup, and ec2:CreatePlacementGroup actions on the database instances (in HANA and NetWeaver on HANA scenarios) in a customer account when they are called via launchwizard.amazonaws.com to enable this support.</td>
<td>April 30, 2021</td>
</tr>
<tr>
<td>AmazonLaunchWizard_Fullaccess</td>
<td>• AWS Launch Wizard added new permissions to create an SNS topic in a customer account, and subscribe to it, unsubscribe from it, and delete it. Permissions are restricted only to resources whose names begin with &quot;Launch Wizard.&quot; AWS Launch Wizard will perform sns:CreateTopic, sns:DeleteTopic, sns:Subscribe, and sns:Unsubscribe actions on a customer account when they are called via launchwizard.amazonaws.com to enable this support.</td>
<td></td>
</tr>
<tr>
<td>AmazonLaunchWizard_Fullaccess</td>
<td>• AWS Launch Wizard added new permissions to perform FSx operations to support SQL Server FCI on AWS Launch Wizard. Launch Wizard will perform fsx:CreateFileSystem, fsx:DescribeFileSystem, fsx:ListTagsForResource, fsx:TagResource, and fsx:UntagResource actions on a customer account when they are called via launchwizard.amazonaws.com to enable this support.</td>
<td></td>
</tr>
<tr>
<td>AmazonLaunchWizard_Fullaccess</td>
<td>• AWS Launch Wizard added a new permission to perform AWS Secrets Manager</td>
<td></td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>operations to support retrieving secret values from Secrets Manager on AWS Launch Wizard</td>
<td>Launch Wizard will perform the <code>arn:aws:secretsmanager:*:secret:LaunchWizard</code> action on a customer account when they are called via <code>launchwizard.amazonaws.com</code> to enable this support.</td>
<td>April 30, 2021</td>
</tr>
<tr>
<td>AWS Launch Wizard started tracking changes</td>
<td>AWS Launch Wizard started tracking changes for its AWS managed policies.</td>
<td></td>
</tr>
</tbody>
</table>
# AWS Launch Wizard documentation history

The following table describes the documentation for this release of AWS Launch Wizard.

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Launch Wizard for Remote Desktop Gateway (p. 44)</td>
<td>You can set up a new Remote Desktop Gateway infrastructure to an existing AWS infrastructure using AWS Launch Wizard for Remote Desktop Gateway.</td>
<td>December 8, 2021</td>
</tr>
<tr>
<td>AWS Launch Wizard for Amazon Elastic Kubernetes Service (p. 26)</td>
<td>Get started setting up a new Amazon EKS application to an existing AWS infrastructure using AWS Launch Wizard.</td>
<td>December 8, 2021</td>
</tr>
<tr>
<td>AWS Launch Wizard for SAP integration with AWS Service Catalog. (p. 66)</td>
<td>You can create AWS Service Catalog products from successful deployments with AWS Launch Wizard.</td>
<td>August 30, 2021</td>
</tr>
<tr>
<td>AWS Launch Wizard for SAP support for no rollback on failure (p. 82)</td>
<td>When you select &quot;No rollback on failure&quot; for your AWS Launch Wizard deployments, if a deployment fails, Launch Wizard does not delete the AWS resources that were created for the deployment.</td>
<td>March 5, 2021</td>
</tr>
<tr>
<td>AWS Launch Wizard for Active Directory support for no rollback on failure (p. 10)</td>
<td>When you select &quot;No rollback on failure&quot; for your AWS Launch Wizard deployments, if a deployment fails, Launch Wizard does not delete the AWS resources that were created for the deployment.</td>
<td>March 5, 2021</td>
</tr>
<tr>
<td>AWS Launch Wizard SQL support for no rollback on failure (p. 167)</td>
<td>When you select &quot;No rollback on failure&quot; for your AWS Launch Wizard deployments, if a deployment fails, Launch Wizard does not delete the AWS resources that were created for the deployment.</td>
<td>March 5, 2021</td>
</tr>
<tr>
<td>AWS Launch Wizard for SAP support for custom IP address specification. (p. 66)</td>
<td>You can specify a private IP address for each Amazon EC2 instance in your SAP deployment.</td>
<td>February 26, 2021</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>SUSE/RHEL high availability for SAP applications (p. 66)</td>
<td>You can configure SUSE/RHEL high availability for SAP applications as part of your deployment with AWS Launch Wizard.</td>
<td>February 1, 2021</td>
</tr>
<tr>
<td>AWS Launch Wizard for SAP support for SAP application installation. (p. 66)</td>
<td>You can install supported SAP applications using customer-provided SAP software.</td>
<td>December 16, 2020</td>
</tr>
<tr>
<td>AWS Launch Wizard for SQL integration with AWS Systems Manager Application Manager. (p. 199)</td>
<td>You can manage resources created by Launch Wizard for SQL from the Systems Manager Application Manager console.</td>
<td>December 15, 2020</td>
</tr>
<tr>
<td>AWS Launch Wizard for Active Directory (p. 1)</td>
<td>You can set up a new Active Directory infrastructure or add domain controllers to an existing AWS infrastructure using AWS Launch Wizard for Active Directory.</td>
<td>December 15, 2020</td>
</tr>
<tr>
<td>AWS Launch Wizard for SAP support for custom pre-deployment and post-deployment scripts. (p. 66)</td>
<td>You can run custom pre- and post-deployment configuration scripts using AWS Launch Wizard for SAP.</td>
<td>November 17, 2020</td>
</tr>
<tr>
<td>AWS Launch Wizard support for SQL Server application single-node deployments. (p. 153)</td>
<td>You can deploy your SQL Server application on a single instance.</td>
<td>October 28, 2020</td>
</tr>
<tr>
<td>AWS Launch Wizard for SAP support for application single-node deployments. (p. 66)</td>
<td>You can deploy your SAP application on a single instance.</td>
<td>October 15, 2020</td>
</tr>
<tr>
<td>Route 53/DNS association support (p. 66)</td>
<td>You can provide your DNS domain name or Route53 hosted zone to enable DNS association for your deployed EC2 instances.</td>
<td>June 18, 2020</td>
</tr>
<tr>
<td>AWS Launch Wizard for SQL Server integration with CloudWatch Application Insights (p. 153)</td>
<td>You can set up monitoring for your application with CloudWatch Application Insights.</td>
<td>June 18, 2020</td>
</tr>
<tr>
<td>SQL Server witness node support. (p. 153)</td>
<td>You can add a witness node to your SQL Server Always On configuration.</td>
<td>May 11, 2020</td>
</tr>
<tr>
<td>Proxy server support (p. 66)</td>
<td>You can route outbound internet traffic for deployed EC2 instances through a proxy server.</td>
<td>May 11, 2020</td>
</tr>
<tr>
<td>Initial release (p. 66)</td>
<td>Initial release of AWS Launch Wizard for SAP User Guide.</td>
<td>April 8, 2020</td>
</tr>
<tr>
<td>Initial release (p. 234)</td>
<td>Initial release of the AWS Launch Wizard for SQL Server User Guide.</td>
<td>November 14, 2019</td>
</tr>
</tbody>
</table>