Contents
Overview ................................................................................................................................. 3
WebSphere Liberty on AWS ................................................................................................. 3
Cost and Licenses .................................................................................................................. 4
AWS Services ....................................................................................................................... 4
Regions, Availability Zones, and Endpoints ......................................................................... 6
Architecture ......................................................................................................................... 7
Deployment Options ........................................................................................................... 11
Deployment Steps ................................................................................................................ 11
Step 1. Prepare an AWS Account ....................................................................................... 12
Step 2. Launch the AWS CloudFormation Template ......................................................... 16
Step 3. Test Your WebSphere Liberty Deployment ............................................................ 22
Troubleshooting .................................................................................................................. 24
Security ................................................................................................................................. 25
AWS Identity and Access Management (IAM) ................................................................. 25
Operating System Security ............................................................................................... 25
Security Groups ................................................................................................................... 25
Additional Resources ......................................................................................................... 26
Git Repository ...................................................................................................................... 27
Document Revisions .......................................................................................................... 27
About This Guide
This Quick Start deployment guide discusses architectural considerations and configuration steps for deploying a highly available architecture for IBM WebSphere Application Server Liberty on the Amazon Web Services (AWS) Cloud. It also provides links for viewing and launching AWS CloudFormation templates that automate the deployment.

The guide is for IT infrastructure architects, administrators, and DevOps professionals who are planning to implement or extend their WebSphere Liberty workloads on the AWS Cloud.

Quick Starts are automated reference deployments for key enterprise workloads on the AWS Cloud. Each Quick Start launches, configures, and runs the AWS compute, network, storage, and other services required to deploy a specific workload on AWS, using AWS best practices for security and availability.

Overview
Cloud is the new normal. Enterprises are rapidly adopting cloud for their businesses to bring in agility, cost-effectiveness, and on-demand scalability.

WebSphere Liberty provides a lightweight, highly composable, and fast-to-start application server environment. It is built on the open-source Open Liberty Project.

WebSphere Liberty on AWS
WebSphere Liberty is a fast, dynamic, and easy-to-use Java application server that combines IBM technology with open-source software. You can use WebSphere Liberty on premises or in the cloud, for development or production. It provides a solid foundation for developing and running service-oriented architecture (SOA) applications on AWS, because of its fast startup times, ability to pick up changes without requiring server restarts, and simple XML configuration.

This guide provides IT infrastructure decision-makers and system administrators with technical guidance on how to configure, deploy, and run WebSphere Liberty in a highly available manner on AWS. It outlines a reference architecture for WebSphere Liberty version 17.0.0.2. This architecture also addresses common scalability, high availability, and security requirements.

This guide discusses best practices for deploying WebSphere Liberty on AWS using Amazon Elastic Compute Cloud (Amazon EC2), Amazon Virtual Private Cloud (Amazon VPC), Application Load Balancer, and Amazon CloudFront. It also provides links to automated
AWS CloudFormation templates that you can launch directly into your AWS account. You can modify these AWS CloudFormation templates to suit your specific business requirements, or use them as is.

WebSphere Liberty has a flexible, modular runtime. After deploying WebSphere Liberty on AWS, you can extend it with additional features as needed. You can also further reduce its footprint for deployment into your production environment. You can integrate Liberty with frameworks like Docker, Chef, Puppet, Jenkins, and UrbanCode Deploy. For further information, join the developer community and see the information on WASdev.net.

You can also migrate your WebSphere-based applications to AWS by using the WebSphere Liberty server, subject to meeting certain criteria. For example, if you’re using the Java EE Web Profile, you can migrate from WebSphere full profile (WAS) to the Liberty Profile (WLP).

For in-depth information about installing and using WebSphere Liberty, see the WebSphere Liberty documentation.

**Cost and Licenses**

This deployment launches WebSphere Liberty 17.0.0.2 automatically into a configuration of your choice. WebSphere Liberty is free for development, small-scale testing, and production use. You are entitled to use up to 2 GB of JVM heap space for running Liberty in test or production. For additional details about licensing and upgrading to a production license that includes support, see the WASdev website. You are responsible for the cost of AWS services used while running this Quick Start reference deployment. However, there is no additional cost for using the Quick Start.

Prices are subject to change. See the pricing pages for each AWS service you will be using for full details. Note that this Quick Start includes nested AWS CloudFormation templates that launch multiple stacks.

**AWS Services**

The core AWS components used by this Quick Start include the following AWS services. (If you are new to AWS, see the Getting Started Resource Center.)

- **Amazon EC2** – The Amazon EC2 service enables you to launch virtual machine instances with a variety of operating systems. You can choose from existing Amazon Machine Images (AMIs) or import your own virtual machine images.
- **Amazon VPC** – The Amazon VPC service lets you provision a private, isolated section of the AWS Cloud where you can launch AWS services and other resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

- **AWS CloudFormation** – AWS CloudFormation gives you an easy way to create and manage a collection of related AWS resources, and provision and update them in an orderly and predictable way. You use a template to describe all the AWS resources (e.g., EC2 instances) that you want. You don’t have to create and configure the resources or figure out dependencies; AWS CloudFormation handles all of that.

- **Auto Scaling** – Auto Scaling helps you maintain high availability and manage capacity by automatically increasing or decreasing the EC2 instance fleet. You can use Auto Scaling to run your fleet at optimal utilization by increasing instance capacity during demand spikes and decreasing capacity during down times.

- **Elastic Load Balancing** – Elastic Load Balancing automatically distributes incoming application traffic across multiple EC2 instances. It detects unhealthy instances and reroutes traffic to healthy instances until the unhealthy instances have been restored. Elastic Load Balancing automatically scales its request handling capacity in response to incoming traffic. Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. This solution uses an Application Load Balancer.

- **IAM** – AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users. With IAM, you can manage users, security credentials such as access keys, and permissions that control which AWS resources users can access, from a central location.

- **Amazon CloudFront** – Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users. CloudFront delivers your content through a worldwide network of data centers called edge locations. CloudFront can be configured with Amazon EC2 or an Application Load Balancer, and can also be configured from external origins such as on-premises data centers.

- **AWS Certificate Manager (ACM)** – This service lets you provision, manage, and deploy Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services. It automatically integrates with a number of AWS services such as Application Load Balancer and CloudFront, and eliminates the need for external certification agencies. SSL/TLS certificates provisioned through ACM are free.
• **Route 53** – Amazon Route 53 is used as a DNS service. It has three functions: registering domain names, routing internet traffic to the resources of your domain, and checking the health of resources. You can use any combination of these functions, or all three.

**Regions, Availability Zones, and Endpoints**

Before you deploy the AWS CloudFormation template, familiarize yourself with AWS Regions, Availability Zones, and endpoints, which are components of the AWS secure global infrastructure.

Use AWS Regions to manage network latency and regulatory compliance. When you store data in a specific region, it isn’t replicated outside that region. It’s your responsibility to replicate data across regions, if your business requires that. AWS provides information about the country, and, where applicable, the state where each region resides; you are responsible for selecting the region to store data, with your compliance and network latency requirements in mind. Regions are designed for high availability and consist of at least two Availability Zones. Most AWS services support regional endpoints, which reduce data latency by providing an entry point for service requests in that region.

Availability Zones are designed for fault isolation. They are connected to multiple internet service providers (ISPs) and different power grids. They are interconnected using high-speed links, so applications can rely on local area network (LAN) connectivity for communication between Availability Zones within the same region. You are responsible for carefully selecting the Availability Zones where your systems will reside. Systems can span multiple Availability Zones, and we recommend that you design your systems to survive temporary or prolonged failure of an Availability Zone in the case of a disaster.

For more information about AWS Regions, Availability Zones, and endpoints, see the [AWS documentation](https://aws.amazon.com/documentation).

AWS provides web access to services through the AWS Management Console, available at [https://aws.amazon.com/console](https://aws.amazon.com/console), and then through individual consoles for each service. AWS provides programmatic access to services through application programming interfaces (APIs) and command line interfaces (CLIs).
Architecture

This Quick Start deploys a production-grade, highly available WebSphere Liberty configuration that you can easily adapt to your use case by customizing parameters.

The Quick Start deployment is automated by nested AWS CloudFormation templates. The main template builds the network-related resources first and then launches a separate template for WebSphere Liberty. Deleting the main template deletes the entire stack.

The Quick Start deploys a cluster of WebSphere Liberty server instances in an Auto Scaling group to provide high availability. An Application Load Balancer distributes the load across the Auto Scaling instances. The architecture also includes an optional Amazon CloudFront distribution for caching static content in edge locations. With the addition of CloudFront, this architecture creates a web layer that prevents direct access to the WebSphere Liberty servers, which are in a private subnet. You can omit CloudFront to decouple edge-caching if desired.

It is also possible to restrict public access so that all requests are routed only through CloudFront and not directly to the Application Load Balancer. For details, see the AWS Security blog post. Figure 1 illustrates the resulting architecture.
Optionally, this Quick Start allows you to create an end-to-end secure architecture, which we recommend. In this case, SSL/TLS certificates are required to be configured at each layer. This architecture uses a self-signed certificate on the WebSphere Liberty server. Certificates provided by ACM are configured to secure the Application Load Balancer and CloudFront layers. Because the certificates are registered against a domain name, alias record sets are created in Amazon Route 53 for both the Application Load Balancer and the CloudFront endpoints. Figure 2 shows the resulting architecture.
The reference architecture deployed by the Quick Start maps AWS services to all the components required by WebSphere Liberty. A highly available reference architecture for WebSphere Liberty requires the following components:

- An HTTP(S) load balancer
- Two or more WebSphere Liberty servers
- A content delivery mechanism that can cache content at the edge
- A DNS system that facilitates routing requests within AWS using alias records

To build this architecture, the Quick Start installs and configures the following:
• A virtual private cloud (VPC) configured across two Availability Zones. In each Availability Zone, this Quick Start provisions one public subnet and one private subnet. This creates a logically isolated networking environment that you can connect to your on-premises data centers or use as a standalone environment.*

• Managed network address translation (NAT) gateways deployed into the public subnets and configured with an Elastic IP address for outbound internet connectivity. These instances provide internet access for all EC2 instances launched within the private network.*

• A Linux bastion host in the public subnet to allow inbound Secure Shell (SSH) access to the WebSphere Liberty instances in the private subnets.*

• In the private subnets, WebSphere Liberty server instances across both Availability Zones, to ensure high availability.

• Auto Scaling enabled for the WebSphere Liberty cluster, to automatically add or remove servers based on their use, providing additional servers during peak hours and lowering costs by removing servers during off hours. This functionality is tightly integrated with the Application Load Balancer, and automatically adds and removes instances from the load balancer. The default installation sets up low and high CPU-based thresholds for scaling the instance capacity up or down. You can modify these thresholds during launch and after deployment.

• The Elastic Load Balancing service, which provides HTTP and HTTPS load balancing across the WebSphere Liberty instances. This Quick Start uses an Application Load Balancer, which is configured to use HTTP.

    Note When you use Elastic Load Balancing, you must upload the web server’s certificate and private key to the IAM service or generate a certificate with ACM before you can enable the HTTPS listener.

• An IAM role with fine-grained permissions for access to AWS services necessary for the deployment process.

• Appropriate security groups for each instance or function to restrict access to only necessary protocols and ports. For example, access to HTTP(S) server ports on Amazon EC2 web servers is limited to the Application Load Balancer.

• Amazon CloudFront as an optional content delivery network. This service enables caching of static content at the edge locations and results in lower latency when delivering content to end users. The edge locations include many Points of Presence (PoPs) across the globe to ensure low latency.
In case of SSL/TLS implementation, AWS Certificate Manager (ACM) to provision certificates for the Application Load Balancer and CloudFront. Alternatively, if you’re using the default CloudFront certificate, ACM is not required.

**Note** This guide doesn’t cover backup procedures. For more information, see the WebSphere Liberty documentation. For additional details, see the Resources section.

Since the certificates are registered against a domain name, in order to route requests, alias records are created for both the Application Load Balancer and CloudFront (if opted) DNS endpoints. Route 53 is used for this purpose.

* The template that deploys the Quick Start into an existing VPC skips the tasks marked by asterisks and prompts you for your existing VPC configuration.

**Deployment Options**

This Quick Start provides two deployment options:

- **Deploy WebSphere Liberty into a new VPC** (end-to-end deployment). This option builds a new AWS environment consisting of the VPC, subnets, NAT gateways, security groups, bastion host, and other infrastructure components, and then deploys WebSphere Liberty into this new VPC.

- **Deploy WebSphere Liberty into an existing VPC**. This option provisions WebSphere Liberty in your existing AWS infrastructure.

**Deployment Steps**

The AWS CloudFormation template provided with this Quick Start bootstraps the AWS infrastructure and automates the deployment of a highly available WebSphere Liberty configuration on the AWS Cloud from scratch. Follow the step-by-step instructions in this section to set up your AWS account, customize the template, and deploy the software into your account.

You can customize the template and the accompanying scripts as needed to best meet your business, IT, and security requirements.

The steps described in this guide will implement a complete WebSphere Liberty environment on the AWS Cloud.
Step 1. Prepare an AWS Account

1. If you don’t already have an AWS account, create one at [https://aws.amazon.com](https://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.

   When you create an AWS account, AWS automatically signs up the account for all AWS services, including Amazon EC2, which you’ll use in the next step. You are charged only for the services that you use.

2. If necessary, request a service limit increase for the EC2 instance types you plan to use for the bastion host and WebSphere Liberty server instances (t2.micro by default) as well as other AWS resources, such as the VPC that you intend to deploy. To do this, in the AWS Support Center, choose **Create Case, Service Limit Increase, EC2 instances** (or VPC), and then complete the fields in the limit increase form, as shown in Figure 3.

   ![Figure 3: Requesting a service limit increase](image)

   **Figure 3: Requesting a service limit increase**

3. Use the region selector in the navigation bar to choose the AWS Region where you want to deploy WebSphere Liberty on AWS.

   Amazon EC2 locations are composed of **AWS Regions** and **Availability Zones**. Regions are dispersed and located in separate geographic areas. This Quick Start uses the t2.micro instance type for the WebSphere Liberty portion of the deployment by default, but you can choose other instance types.
Note  This Quick Start is supported in all the regions shown in Figure 4.

Figure 4: Choosing an AWS Region

Tip  Consider choosing a region closest to your data center or corporate network to reduce network latency between systems running on AWS and the systems and users on your corporate network.

4. Create a key pair in your preferred region. To do this, in the navigation pane of the Amazon EC2 console, choose Key Pairs, Create Key Pair, type a name, and then choose Create. Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. On Linux, the key pair is used to authenticate SSH login. Skip this step if you already have a key pair you can use.
5. If you’re planning to use SSL/TLS in your architecture, follow these additional steps:
   a. Create valid certificates for the Application Load Balancer and CloudFront (if opted), using ACM in the same region. To request a certificate, follow the instructions in the ACM documentation.

   **Note**  All certificates used in this architecture should have a common domain name; for example, alb.wasliberty.qs.com and cloudfront.wasliberty.qs.com. We recommend that you register a single certificate with a wildcard and use it for both the Application Load Balancer and CloudFront; for example, *.wasliberty.qs.com, as shown in Figure 6.
b. Create a valid hosted zone that matches the domain name of the certificate in Route 53, as shown in Figure 7. For instructions, see the Route 53 documentation.
Also, note that CloudFront can be configured to use its own default certificate. In this case, an ACM Certificate is not required.

Skip this step if you aren’t using SSL/TLS.

**Step 2. Launch the AWS CloudFormation Template**

**Note** You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using this Quick Start. For full details, see the pricing pages for each AWS service you will be using in this Quick Start. Prices are subject to change.

1. Choose one of the following options to launch the AWS CloudFormation template into your AWS account. For help choosing an option, see deployment options earlier in this guide.

<table>
<thead>
<tr>
<th>Option 1</th>
<th>Option 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deploy WebSphere Liberty into a new VPC</td>
<td>Deploy WebSphere Liberty into an existing VPC</td>
</tr>
</tbody>
</table>

**Important** If you’re deploying WebSphere Liberty into an existing VPC, make sure that your VPC has two private subnets in different Availability Zones for WebSphere Liberty instances. These subnets require NAT gateways in their route tables, to allow the instances to download packages and software without exposing them to the internet. You’ll also need the domain name option configured in the DHCP options, as explained in the Amazon VPC documentation. You will be prompted for your VPC settings when you launch the Quick Start.

Each template is launched in the US East (N. Virginia) Region by default. You can change the region by using the region selector in the navigation bar.

Each stack takes approximately 40 minutes to create.

2. On the Select Template page, keep the default setting for the template URL, and then choose Next.
3. On the **Specify Details** page, review the parameters for the template. Provide values for the parameters that require your input. For all other parameters, the template provides default settings that you can customize.

In the following tables, parameters are listed by category and described separately for the two deployment options:

- **Parameters for deploying WebSphere Liberty into a new VPC**
- **Parameters for deploying WebSphere Liberty into an existing VPC**

**Option 1: Parameters for deploying WebSphere Liberty into a new VPC**

**View template**

**Network Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability Zones (AvailabilityZones)</td>
<td><strong>Requires input</strong></td>
<td>The list of Availability Zones to use for the subnets in the VPC. The Quick Start uses two Availability Zones from your list and preserves the logical order you specify.</td>
</tr>
<tr>
<td>VPC CIDR (VPCCIDR)</td>
<td>10.0.0.0/16</td>
<td>The CIDR block for the VPC.</td>
</tr>
<tr>
<td>Private Subnet 1 CIDR (PrivateSubnet1CIDR)</td>
<td>10.0.0.0/19</td>
<td>The CIDR block for the private subnet located in Availability Zone 1.</td>
</tr>
<tr>
<td>Private Subnet 2 CIDR (PrivateSubnet2CIDR)</td>
<td>10.0.32.0/19</td>
<td>The CIDR block for the private subnet located in Availability Zone 2.</td>
</tr>
<tr>
<td>Public Subnet 1 CIDR (PublicSubnet1CIDR)</td>
<td>10.0.128.0/20</td>
<td>The CIDR block for the public (DMZ) subnet located in Availability Zone 1.</td>
</tr>
<tr>
<td>Public Subnet 2 CIDR (PublicSubnet2CIDR)</td>
<td>10.0.144.0/20</td>
<td>The CIDR block for the public (DMZ) subnet located in Availability Zone 2.</td>
</tr>
<tr>
<td>Allowed Bastion External Access CIDR (RemoteAccessCIDR)</td>
<td><strong>Requires input</strong></td>
<td>The CIDR IP range that is permitted to access the WebSphere Liberty software. We recommend that you set this value to a trusted IP range. For example, you might want to grant only your corporate network access to the software.</td>
</tr>
</tbody>
</table>

**Bastion Host Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Pair Name (KeyPairName)</td>
<td><strong>Requires input</strong></td>
<td>A public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region.</td>
</tr>
<tr>
<td>Parameter label (name)</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>Bastion AMI Operating System (BastionAMIOS)</td>
<td>Amazon-Linux-HVM</td>
<td>The Linux operating system for the bastion host instance. If you choose CentOS, make sure that you have a subscription to the <a href="https://aws.amazon.com/marketplace">CentOS AMI in AWS Marketplace</a>.</td>
</tr>
<tr>
<td>Bastion Instance Type (BastionInstanceType)</td>
<td>t2.micro</td>
<td>The EC2 instance type for the bastion host instance.</td>
</tr>
</tbody>
</table>

**Liberty Nodes Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liberty Server Instance Type (LibertyInstanceType)</td>
<td>t2.micro</td>
<td>The EC2 instance type for the WebSphere Liberty instances.</td>
</tr>
<tr>
<td>Liberty Nodes Min Size (LibertyNodesMinSize)</td>
<td>2</td>
<td>The minimum size of the Auto Scaling group for the WebSphere Liberty instances.</td>
</tr>
<tr>
<td>Liberty Nodes Max Size (LibertyNodesMaxSize)</td>
<td>4</td>
<td>The maximum size of the Auto Scaling group for the WebSphere Liberty instances.</td>
</tr>
<tr>
<td>Liberty Nodes Desired Capacity (LibertyNodesDesiredCapacity)</td>
<td>2</td>
<td>The desired capacity of the Auto Scaling group for the WebSphere Liberty instances.</td>
</tr>
<tr>
<td>Operator Email (OperatorEmail)</td>
<td>Requires input</td>
<td>The email that notifications will be sent to when the environment is scaled up or down.</td>
</tr>
<tr>
<td>Deploy Sample Application? (DeploySampleApp)</td>
<td>No</td>
<td>Set this parameter to Yes if you want to download and deploy the <a href="https://github.com/IBM/Ferre">Ferre sample application</a> from the WASdev repository.</td>
</tr>
</tbody>
</table>

**SSL Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARN of ACM Certificate for Application Load Balancer (ALBSSLCertificateARN)</td>
<td>Optional</td>
<td>The ARN of the SSL certificate to be used for the Application Load Balancer. This parameter is mandatory for SSL/TLS implementation.</td>
</tr>
<tr>
<td>ARN of ACM Certificate for CloudFront (CloudFrontACMCertificateARN)</td>
<td>Optional</td>
<td>The ARN of the ACM Certificate for the CloudFront distribution. This certificate should be created in the us-east-1 (N. Virginia) Region and must reference the main domain name you use for the hosted zone. Leave blank if SSL/TLS is not required or if you opt to use the default CloudFront certificate.</td>
</tr>
<tr>
<td>Hosted Zone Name in Route 53 corresponding to the domain name of the web site (HostedZoneName)</td>
<td>Optional</td>
<td>The domain name of a valid hosted zone on AWS, e.g., example.com. This parameter is mandatory for SSL/TLS implementation. Please follow the formatting requirements in the AWS documentation.</td>
</tr>
<tr>
<td>Parameter label (name)</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Create a CloudFront Distribution?</td>
<td>Yes</td>
<td>Set this parameter to No if you don’t want to create a CloudFront distribution that will cache content at the edge location. By default, the distribution is configured with the Application Load Balancer endpoint as the origin domain.</td>
</tr>
<tr>
<td>CloudFront CNAME Alias</td>
<td>Required if previous parameter is set to Yes</td>
<td>The alternate domain name for the CloudFront distribution (e.g., my-liberty-cf.example.com). This alias must match the specified hosted zone name. This parameter is required if the previous parameter (Create a CloudFront Distribution?) is set to Yes. Please follow the formatting requirements in the AWS documentation.</td>
</tr>
</tbody>
</table>

**AWS Quick Start Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick Start S3 Bucket Name</td>
<td>aws-quickstart</td>
<td>The S3 bucket you have created for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. The bucket name can include numbers, lowercase letters, uppercase letters, and hyphens, but should not start or end with a hyphen.</td>
</tr>
<tr>
<td>Quick Start S3 Key Prefix</td>
<td>quickstart-ibm-websphere-liberty/</td>
<td>The S3 key name prefix used to simulate a folder for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. This prefix can include numbers, lowercase letters, uppercase letters, hyphens, and forward slashes.</td>
</tr>
</tbody>
</table>

- **Option 2: Parameters for deploying WebSphere Liberty into an existing VPC**

  View template

**Network Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPC ID (VPCID)</td>
<td>Requires input</td>
<td>The ID of your existing VPC (e.g., vpc-0343606e).</td>
</tr>
<tr>
<td>Private Subnet 1 ID (PrivateSubnet1ID)</td>
<td>Requires input</td>
<td>The ID of the private subnet in Availability Zone 1 in your existing VPC, for the WebSphere Liberty instances (e.g., subnet-a0246dcd).</td>
</tr>
<tr>
<td>Private Subnet 2 ID (PrivateSubnet2ID)</td>
<td>Requires input</td>
<td>The ID of the private subnet in Availability Zone 2 in your existing VPC, for the WebSphere Liberty instances (e.g., subnet-b1f432cd).</td>
</tr>
</tbody>
</table>
### Parameter label (name) | Default | Description
---|---|---
**Public Subnet 1 ID** *(PublicSubnet1ID)* | *Requires input* | The ID of the public subnet in Availability Zone 1 in your existing VPC, for the Application Load Balancer (e.g., subnet-9fc642ac).  
**Public Subnet 2 ID** *(PublicSubnet2ID)* | *Requires input* | The ID of the public subnet in Availability Zone 2 in your existing VPC, for the Application Load Balancer (e.g., subnet-e3246d8e).

### Bastion Host Details:

**Parameter label (name) | Default | Description
---|---|---
**SSH Key Name** *(KeyPairName)* | *Requires input* | A public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region.  
**Bastion Security Group ID** *(BastionSecurityGroupID)* | *Requires input* | The ID of the bastion host security group, to enable SSH connections (e.g., sg-7f16e910).

### Liberty Nodes Configuration:

**Parameter label (name) | Default | Description
---|---|---
**Liberty Server Instance Type** *(LibertyInstanceType)* | t2.micro | The EC2 instance type for the WebSphere Liberty instances.  
**Liberty Nodes Min Size** *(WebServerMinSize)* | 2 | The minimum size of the Auto Scaling group for the WebSphere Liberty instances.  
**Liberty Nodes Max Size** *(LibertyNodesMaxSize)* | 4 | The maximum size of the Auto Scaling group for the WebSphere Liberty instances.  
**Liberty Nodes Desired Capacity** *(LibertyNodesDesiredCapacity)* | 2 | The desired capacity of the Auto Scaling group for the WebSphere Liberty instances.  
**Operator Email** *(OperatorEmail)* | *Requires input* | The email that notifications will be sent to when the environment is scaled up or down.  
**Deploy Sample Application?** *(DeploySampleApp)* | No | Set this parameter to Yes if you want to download and deploy the Ferret sample application from the WASdev repository.

### SSL Configuration:

**Parameter label (name) | Default | Description
---|---|---
**ARN of ACM Certificate for Application Load Balancer** *(ALBSSLCertificateARN)* | *Optional* | The ARN of the SSL certificate to be used for the Application Load Balancer. This parameter is mandatory for SSL/TLS implementation.
<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARN of ACM Certificate for CloudFront (CloudFrontACMCertificate ARN)</td>
<td>Optional</td>
<td>The ARN of the ACM Certificate for the CloudFront distribution. This certificate should be created in the us-east-1 (N. Virginia) Region and must reference the main domain name you use for the hosted zone. Leave blank if SSL/TLS is not required or if you opt to use the default CloudFront certificate.</td>
</tr>
<tr>
<td>Hosted Zone Name in Route 53 corresponding to the domain name of the website (HostedZoneName)</td>
<td>Optional</td>
<td>The domain name of a valid hosted zone on AWS; e.g., example.com. This parameter is mandatory for SSL/TLS implementation. Please follow the formatting requirements in the AWS documentation.</td>
</tr>
<tr>
<td>Create a CloudFront Distribution? (CreateCloudFrontDistribution)</td>
<td>Yes</td>
<td>Set this parameter to No if you don’t want to create a CloudFront distribution that will cache content at the edge location. By default, the distribution is configured with the Application Load Balancer endpoint as the origin domain.</td>
</tr>
<tr>
<td>CloudFront CNAME Alias (LibertyCloudFrontCNAME)</td>
<td>Required if previous parameter is set to Yes</td>
<td>The alternate domain name for the CloudFront distribution (e.g., my-liberty-cf.example.com). This alias must match the specified hosted zone name. This parameter is required if the previous parameter (Create a CloudFront Distribution?) is set to Yes. Please follow the formatting requirements in the AWS documentation.</td>
</tr>
</tbody>
</table>

**AWS Quick Start Configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick Start S3 Bucket Name (QSS3BucketName)</td>
<td>aws-quickstart</td>
<td>The S3 bucket you have created for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. The bucket name can include numbers, lowercase letters, uppercase letters, and hyphens, but should not start or end with a hyphen.</td>
</tr>
<tr>
<td>Quick Start S3 Key Prefix (QSS3KeyPrefix)</td>
<td>quickstart-ibm-websphere-liberty/</td>
<td>The S3 key name prefix used to simulate a folder for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. This prefix can include numbers, lowercase letters, uppercase letters, hyphens, and forward slashes.</td>
</tr>
</tbody>
</table>

4. On the **Options** page, you can specify tags (key-value pairs) for resources in your stack and set advanced options. When you’re done, choose **Next**.

5. On the **Review** page, review and confirm the template settings. Under **Capabilities**, select the check box to acknowledge that the template will create IAM resources.
6. Choose **Create** to deploy the stack.
7. Monitor the status of the stack. When the status is **CREATE_COMPLETE**, the WebSphere Liberty servers are ready. Depending on the instance type you chose, you might have to wait for a few minutes for the cluster to be ready.

This completes the creation of the WebSphere Liberty environment on AWS.

**Step 3. Test Your WebSphere Liberty Deployment**

In the [AWS CloudFormation console](https://console.aws.amazon.com/cloudformation/home), in the **Outputs** tab, choose the link for the **WebsiteURL** key, as shown in Figure 8.

![Figure 8: AWS CloudFormation output link](https://console.aws.amazon.com/cloudformation/home)

This link launches the WebSphere Liberty console, which is the web client interface for WebSphere Liberty administration, as shown in Figure 9. Note that the link points to the output of the Quick Start that uses custom ACM Certificates for CloudFront and the Application Load Balancer.
If you set the **Deploy Sample Application?** parameter to **Yes**, the Quick Start also deploys the [Ferret sample application](https://github.com/IBM/ibm-wa-rxds-examples) from the WASdev repository. Figure 10 shows the sample application page rendered in the browser.
Now that you have validated the deployment, you can follow the instructions in the IBM documentation to use WebSphere Liberty.

**Troubleshooting**

**Q.** I encountered a CREATE_FAILED error when I launched the Quick Start.

**A.** If AWS CloudFormation fails to create the stack, we recommend that you relaunch the template with **Rollback on failure** set to **No**. (This setting is under **Advanced** in the AWS CloudFormation console, **Options** page.) With this setting, the stack's state will be retained and the instance will be left running, so you can troubleshoot the issue. (Look at the log files in `%ProgramFiles%\Amazon\EC2ConfigService` and `C:\cfn\log`.)

**Important** When you set **Rollback on failure** to **No**, you will continue to incur AWS charges for this stack. Please make sure to delete the stack when you finish troubleshooting.

You might also get a CREATE_FAILED error if you enter an uppercase string for the **CloudFront CNAME Alias** parameter. The alias forms a part of the CNAME for
CloudFront, and CNAMEs must be lowercase. Set the alias to lowercase letters and relaunch the Quick Start.

Security

The AWS Cloud provides a scalable, highly reliable platform that helps customers deploy applications and data quickly and securely. When you build systems on the AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. In turn, you assume responsibility and management of the guest operating system (including updates and security patches) and other associated applications, as well as the configuration of the AWS-provided security group firewall. For more information about security on AWS, visit the AWS Security Center.

AWS Identity and Access Management (IAM)

This solution uses an IAM role with least privileged access. We do not require or recommend storing SSH keys, secret keys, or access keys on the provisioned instances.

When you launch the Quick Start, if you select the check box to acknowledge that the template will create IAM resources (under Capabilities), AWS CloudFormation will automatically acquire the IAM resources.

Operating System Security

The root user on cluster nodes can be accessed only by using the SSH key specified during the deployment process. AWS doesn’t store these SSH keys, so if you lose your SSH key you can lose access to these instances.

Operating system patches are your responsibility and should be performed on a periodic basis.

Security Groups

A security group acts as a firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time. The new rules are automatically applied to all instances that are associated with the security group.

This Quick Start deployment creates the following security groups:
- Application Load Balancer
- WebSphere Liberty security group
- NAT gateways and bastion hosts

### Additional Resources

**AWS services**

- AWS CloudFormation
- Amazon EC2
- Amazon VPC
- Auto Scaling
- Application Load Balancer
- Amazon S3
- Amazon IAM
- Amazon CloudFront
- AWS Certificate Manager
- Route 53

**WebSphere Liberty**

- WebSphere Liberty
  [https://www.ibm.com/support/knowledgecenter/SSEQTP_liberty/as_ditamaps/was90o>Welcome>Ledelty.html](https://www.ibm.com/support/knowledgecenter/SSEQTP_liberty/as_ditamaps/was90o>Welcome>Ledelty.html)
• WebSphere Liberty Profile
  http://www.adam-bien.com/roller/abien-entry/small_footprint_fast_deployments_easy

• WebSphere Liberty on the cloud

Quick Start Reference Deployments

• AWS Quick Start home page
  https://aws.amazon.com/quickstart/

GitHub Repository

You can visit our GitHub repository to download the templates and scripts for this Quick Start, to post your feedback, and to share your customizations with others.

Document Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
<th>In sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 2018</td>
<td>Initial publication</td>
<td>–</td>
</tr>
</tbody>
</table>
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Notices

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

The software included with this paper is licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with the License. A copy of the License is located at http://aws.amazon.com/apache2.0/ or in the "license" file accompanying this file. This code is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.