

WordPress High Availability by Bitnami on the AWS Cloud

Quick Start Reference Deployment

September 2018

([last update](#): January 2019)

*Andres Bono Jimenez, Raquel Campuzano, Tomas Pizarro Moreno,
and Rafael Rios Saavedra, Bitnami
Andrew Glenn, AWS Quick Start Team*

Contents

Quick Links	2
Overview.....	3
WordPress High Availability by Bitnami on AWS.....	3
Costs and Licenses.....	4
Architecture.....	5
Prerequisites	6
Specialized Knowledge	6
Deployment Options	7
Deployment Steps	7
Step 1. Prepare Your AWS Account.....	7
Step 2. Launch the Quick Start	8
Step 3. Test the Deployment	21
Step 4. (Optional) Test the W3 Total Cache Plugin	24
Step 5. (Optional) Configure OPcache	25
Best Practices Using WordPress High Availability by Bitnami on AWS	27

WordPress Updates	27
Updating WordPress Components Manually	28
Updating WordPress from the Dashboard	28
Getting Major Release Updates from Bitnami.....	28
Security.....	28
AWS Identity and Access Management (IAM)	29
Operating System Security	29
Other Security Considerations	29
FAQ.....	30
Bitnami Support for AWS	31
GitHub Repository	31
Additional Resources	31
Document Revisions	32

This Quick Start was created by Bitnami in collaboration with Amazon Web Services (AWS).

[Quick Starts](#) are automated reference deployments that use AWS CloudFormation templates to deploy key technologies on AWS, following AWS best practices.

Quick Links

The links in this section are for your convenience. Before you launch the Quick Start, please review the architecture, security, and other considerations discussed in this guide.

- If you have an AWS account, and you're already familiar with AWS services and WordPress High Availability by Bitnami, you can launch the Quick Start to build the architecture shown in [Figure 1](#) in a new or existing virtual private cloud (VPC). The deployment takes approximately 40 minutes. If you're new to AWS or to WordPress High Availability by Bitnami, please review the implementation details and follow the [step-by-step instructions](#) provided later in this guide.

Launch
(for new VPC)

Launch
(for existing VPC)

- If you want to take a look under the covers, you can view the AWS CloudFormation templates that automate the deployment.

View template
(for new VPC)

View template
(for existing VPC)

Overview

This Quick Start reference deployment guide provides step-by-step instructions for deploying WordPress High Availability by Bitnami on the AWS Cloud.

This Quick Start is for users who are planning to implement or extend their WordPress workloads on the AWS Cloud, including IT infrastructure architects, administrators, and DevOps professionals.

WordPress High Availability by Bitnami on AWS

WordPress is a web publishing platform for building blogs and websites. It can be customized via a wide selection of themes, extensions, and plugins.

WordPress provides a development framework, extensive feature set, flexibility, rapid and multilingual publishing ability, multi-author support, and thriving community. For more information about these features, see the [WordPress website](#).

WordPress High Availability by Bitnami enables you to deploy the WordPress application and database on different Amazon Elastic Compute Cloud (Amazon EC2) instances. The Quick Start uses AWS Relational Database Service (Amazon RDS) with Amazon Aurora to set up a relational database in the AWS Cloud that helps you reduce costs, simplify configuration tasks, and scale with ease. Optionally, you can also deploy an Amazon ElastiCache for Memcached server to cache database queries.

This Quick Start implementation of WordPress High Availability by Bitnami provides benefits such as the following:

High performance

- WordPress workloads are deployed on multiple servers (EC2 instances) for better performance.
- You can add capacity by increasing the number of nodes in the cluster.

- You can improve the WordPress cache performance by configuring the W3 Total Cache plugin to use the ElastiCache server.
- PHP's byte code cache OPcache is enabled by default.

High availability and failover

- If a node is down, the cluster can continue working.

Security

- The Aurora database and WordPress application (data and code) are provisioned on separate EC2 instances to help improve security and access control.
- The WordPress environment is deployed into a virtual private cloud (VPC) that includes bastion hosts and stack instances to provide secure access to private subnets.
- Users can access the server through Secure Shell (SSH) by connecting to the bastion host instance.

Additional features

- Log rotation and system monitoring ([Gonit](#)) are included by default.

Costs and Licenses

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 the Quick Start.

The AWS CloudFormation template for this Quick Start includes configuration parameters that you can customize. Some of these settings, such as instance type, will affect the cost of deployment. For cost estimates, see the pricing pages for each AWS service you will be using. Prices are subject to change.

Tip After you deploy the Quick Start, we recommend that you enable the [AWS Cost and Usage Report](#) to track costs associated with the Quick Start. This report delivers billing metrics to an S3 bucket in your account. It provides cost estimates based on usage throughout each month, and finalizes the data at the end of the month. For more information about the report, see the [AWS documentation](#).

WordPress High Availability by Bitnami and its components are provided as open-source software, and are distributed under the following licenses:

- WordPress: [GNU General Public License version 2](#) (GPL2)

- Apache HTTP Server: [Apache License version 2.0](#) (APACHE2)
- PHP: [The PHP License version 3.01](#) (PHP)
- MySQL client: [GNU General Public License version 2](#) (GPL2)
- Gonic: [GNU General Public License version 2](#) (GPL2)

To check licenses for other components deployed by this Quick Start, view the /opt/bitnami/licenses folder of your stack after you deploy the Quick Start.

Architecture

Deploying this Quick Start for a new VPC with **default parameters** builds the following WordPress High Availability by Bitnami environment in the AWS Cloud.

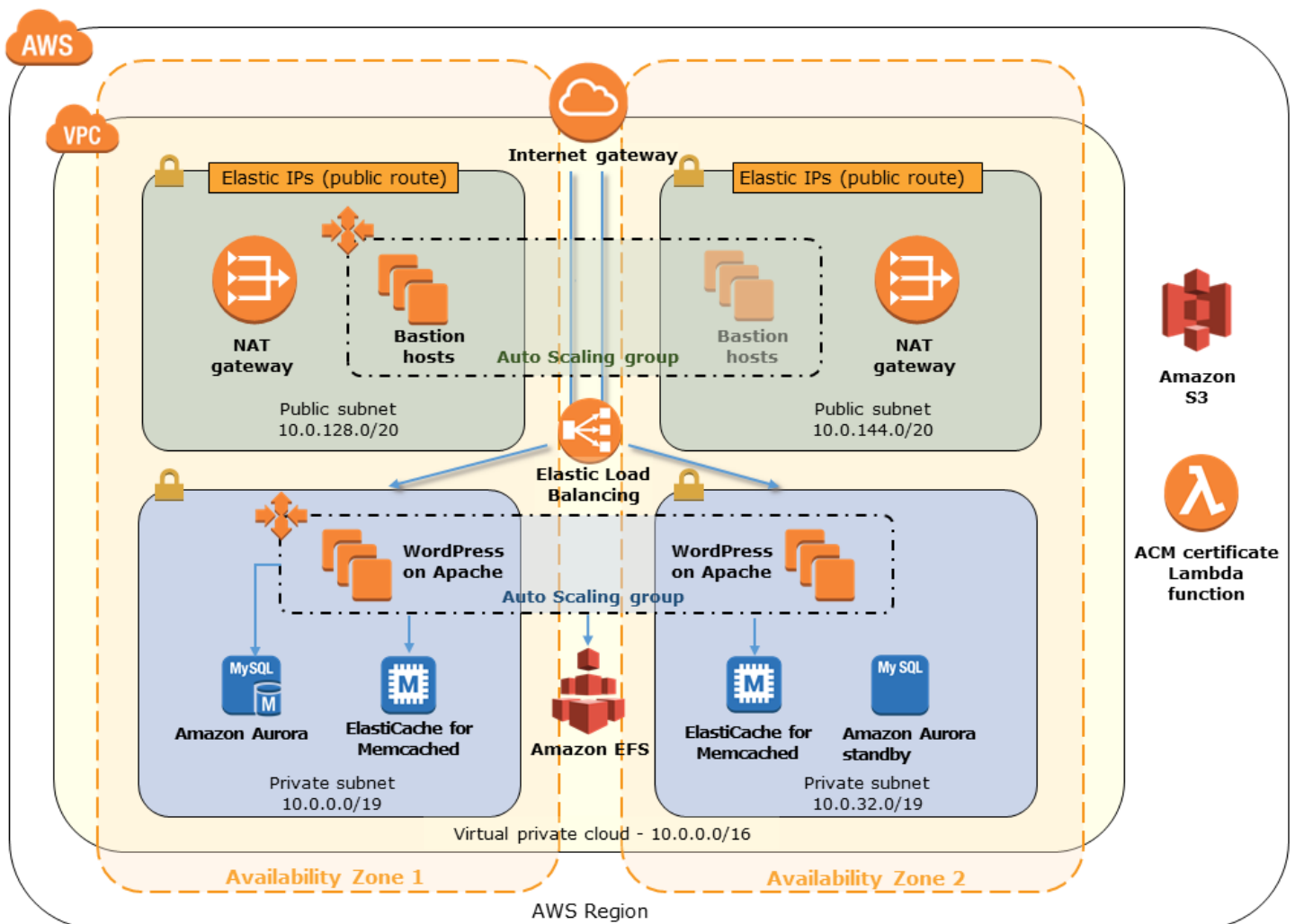


Figure 1: Quick Start architecture for WordPress High Availability by Bitnami

The Quick Start sets up the following:

- A highly available architecture that spans two Availability Zones.*
- 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 bastion hosts to send and receive traffic.*
- In the public subnets, managed NAT gateways to allow outbound internet access for resources in the private subnets.*
- In the public subnets, Linux bastion hosts in an Auto Scaling group to allow inbound Secure Shell (SSH) access to EC2 instances in public and private subnets.*
- Elastic Load Balancing (ELB) to distribute HTTP and HTTPS requests across multiple WordPress instances.
- In the private subnets, EC2 instances that host the WordPress application on Apache. These instances are provisioned in an Auto Scaling group to ensure high availability.
- In the private subnets, Aurora DB instances administered by Amazon RDS.
- In the private subnets, Amazon Elastic File System (Amazon EFS) to share assets (such as plugins, themes, and images) across WordPress instances.
- In the private subnets, Amazon ElastiCache for Memcached nodes for caching database queries.

* 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.

Prerequisites

Specialized Knowledge

Before you deploy this Quick Start, we recommend that you become familiar with the following AWS services. (If you are new to AWS, see [Getting Started with AWS](#).)

- [AWS CloudFormation](#)
- [Amazon Aurora](#)
- [Amazon EBS](#)
- [Amazon EC2](#)
- [Amazon EC2 Auto Scaling](#)
- [Amazon EFS](#)

- [Amazon ElastiCache](#)
- [Amazon RDS](#)
- [Amazon VPC](#)
- [Elastic Load Balancing](#)

Deployment Options

This Quick Start provides two deployment options:

- **Deploy WordPress High Availability by Bitnami 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 hosts, and other infrastructure components, and then deploys WordPress High Availability by Bitnami into this new VPC.
- **Deploy WordPress High Availability by Bitnami into an existing VPC.** This option provisions WordPress High Availability by Bitnami in your existing AWS infrastructure.

The Quick Start provides separate templates for these options. It also lets you configure CIDR blocks, instance types, and WordPress High Availability by Bitnami settings, as discussed later in this guide.

Deployment Steps

Step 1. Prepare Your AWS Account

1. If you don't already have an AWS account, create one at <https://aws.amazon.com> by following the on-screen instructions.
2. Use the region selector in the navigation bar to choose the AWS Region where you want to deploy WordPress High Availability by Bitnami on AWS.

Important This Quick Start includes Amazon EFS, which isn't available in all AWS Regions. For a list of supported regions, see [AWS Regions and Endpoints](#).

3. Create a [key pair](#) in your preferred region.

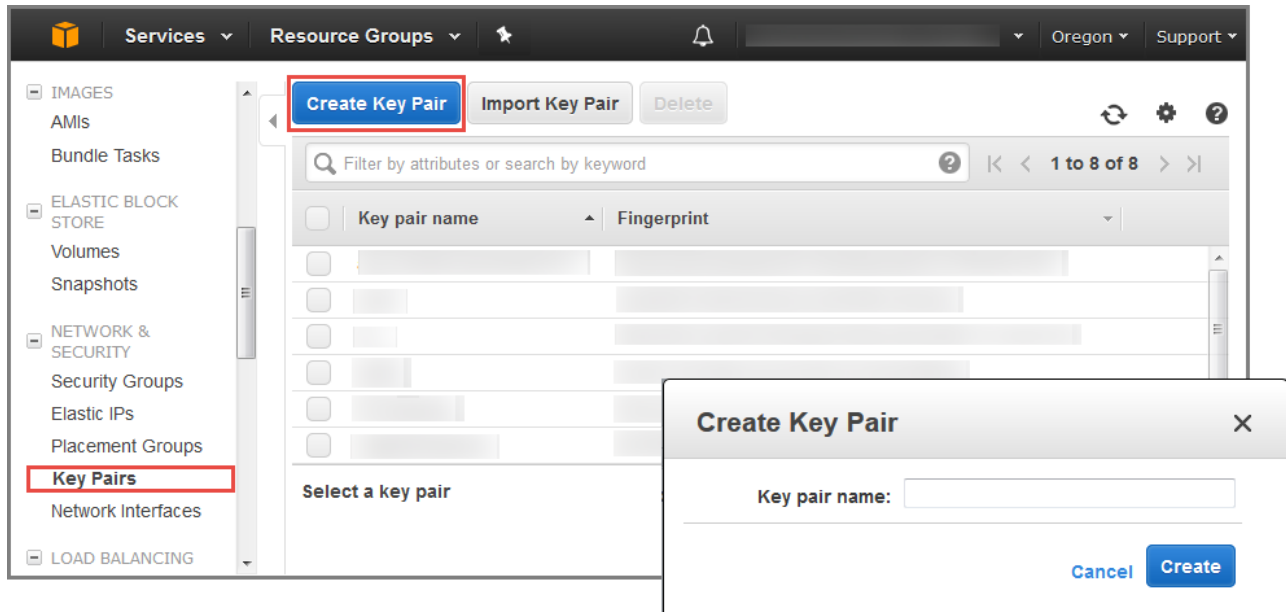


Figure 2: Creating a key pair

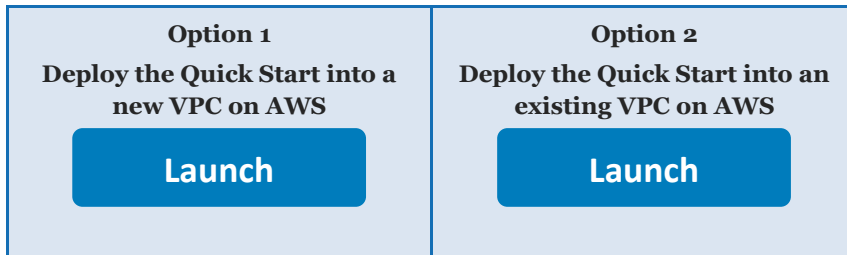
Note You will be able to download the private key file only once. Store it safely; you won't be able to log in to your AWS instances without it.

4. If necessary, request a service quota increase for the instance types used for the deployment. You might need to request an increase if you need additional Elastic IP addresses or if you already have an existing deployment that uses the same instance types as this architecture. To do this, on the [Service Quotas](#) console, for each instance type that you want a service quota increase, choose the instance type, choose **Request quota increase**, and then complete the fields in the quota increase form. It can take a few days for the new service quota to become effective.

Step 2. Launch the Quick Start

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. If you are using the CentOS operating system, subscribe to the [CentOS AMI in AWS Marketplace](#).
2. 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.



Important If you're deploying WordPress High Availability by Bitnami into an existing VPC, make sure that your VPC has two private subnets in different Availability Zones for the database instances. These subnets require [NAT gateways or NAT instances](#) in their route tables, to allow the instances to download packages and software without exposing them to the internet. You will 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 deployment takes about 40 minutes to complete.

3. Check the region that's displayed in the upper-right corner of the navigation bar, and change it if necessary. This is where the network infrastructure for WordPress High Availability by Bitnami will be built. The template is launched in the US East (Ohio) Region by default.

Important This Quick Start uses Amazon EFS, which isn't available in all AWS Regions. For a list of supported regions, see [AWS Regions and Endpoints](#).

4. On the **Select Template** page, keep the default setting for the template URL, and then choose **Next**.

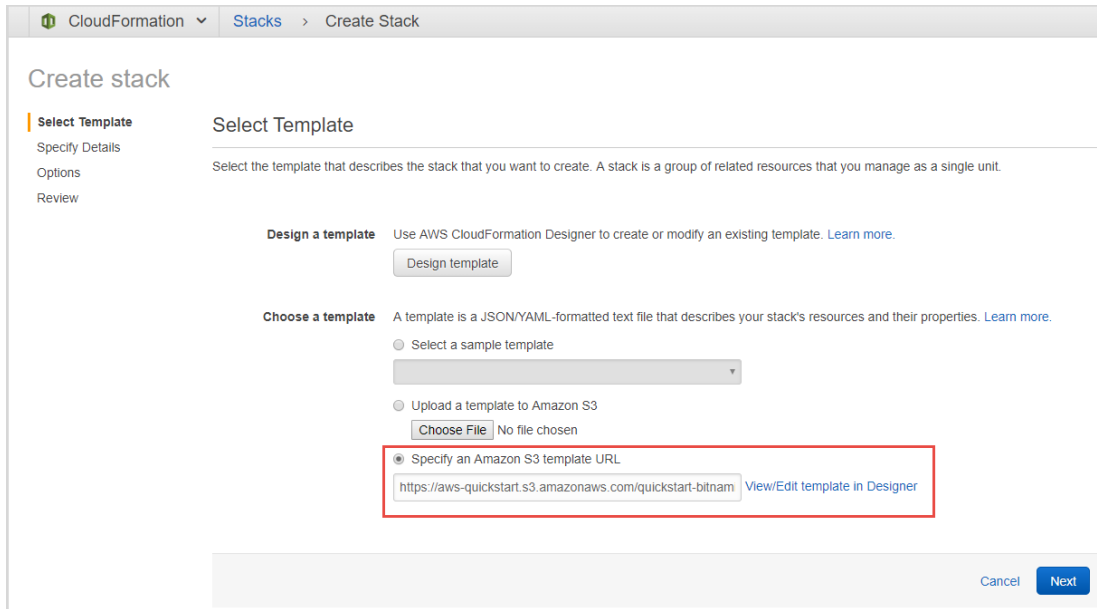


Figure 3: Quick Start template location

5. On the **Specify Details** page, change the stack name if needed. Review the parameters for the template. Provide values for the parameters that require input. For all other parameters, review the default settings and customize them as necessary. When you finish reviewing and customizing the parameters, choose **Next**.

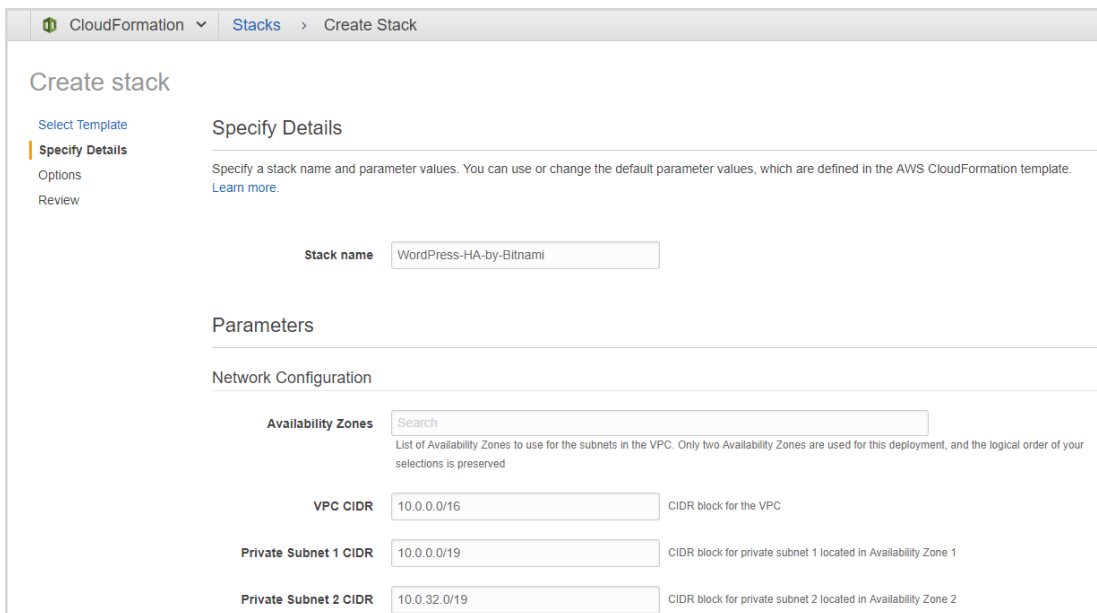


Figure 4: Completing the Quick Start parameters

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

- [Parameters for deploying WordPress High Availability by Bitnami into a new VPC](#)
 - [Parameters for deploying WordPress High Availability by Bitnami into an existing VPC](#)
- **Option 1: Parameters for deploying WordPress High Availability by Bitnami into a new VPC**

[View template](#)

VPC Network Configuration:

Parameter label (name)	Default	Description
Availability Zones (AvailabilityZones)	<i>Requires input</i>	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.
VPC CIDR (VPCCIDR)	10.0.0.0/16	The CIDR block for the VPC.
Private Subnet 1 CIDR (PrivateSubnet1CIDR)	10.0.0.0/19	The CIDR block for the private subnet located in Availability Zone 1.
Private Subnet 2 CIDR (PrivateSubnet2CIDR)	10.0.32.0/19	The CIDR block for the private subnet located in Availability Zone 2.
Public Subnet 1 CIDR (PublicSubnet1CIDR)	10.0.128.0/20	The CIDR block for the public (DMZ) subnet located in Availability Zone 1.
Public Subnet 2 CIDR (PublicSubnet2CIDR)	10.0.144.0/20	The CIDR block for the public (DMZ) subnet located in Availability Zone 2.
Allowed CIDR for ALB Access (ALBAccessCIDR)	10.0.0.0/16	The CIDR IP range that is permitted external web access to the Application Load Balancer. We recommend that you set this value to a trusted IP range.

Linux Bastion Configuration:

Parameter label (name)	Default	Description
Bastion Instance Type (BastionInstanceType)	t2.micro	The Amazon EC2 instance type for the bastion host instances.
Bastion AMI OS (BastionAMIOS)	Amazon-Linux-HVM	The Linux distribution for the AMI to be used for the bastion host instances. If you choose CentOS, make sure that you have a subscription to the CentOS AMI in AWS Marketplace .

Parameter label (name)	Default	Description
Allowed Bastion External Access CIDR (RemoteAccessCIDR)	127.0.0.1/32	The CIDR IP range that is permitted external SSH access to the bastion host instances. We recommend that you set this value to a trusted IP range.
SSH KeyPair Name (KeyPairName)	<i>Requires input</i>	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.

Amazon RDS (Aurora) Configuration:

Parameter label (name)	Default	Description
Enable Auto Minor Version Upgrade (DBAutoMinorVersionUpgrade)	true	Determines whether the DB instance will automatically be upgraded to new Aurora or MySQL minor versions as they are supported by Amazon RDS. Set this parameter to false to disable auto-upgrades.
Backup Retention Period (DBBackupRetentionPeriod)	7	Number of days for which automatic DB snapshots are retained. You can set this parameter to a value between 1 and 35 days.
Preferred Backup Window (DBPreferredBackupWindow)	<i>Optional</i>	The preferred time period for automated backups, specified in the format HH:MM-HH:MM (daily start time-daily end time), using coordinated universal time (UTC). For example, 09:15-13:15 would set the start time to 9:15 AM UTC, and the end time to 1:15 PM UTC. Backups that are created during this window are retained for the number of days specified by the Backup Retention Period parameter. For more information, see the Amazon RDS documentation .
Database Instance Size (DBInstanceClass)	db.t2.small	The instance size (compute and memory capacity) class for the Aurora DB instances.
Database Admin Password (DBMasterUserPassword)	<i>Requires input</i>	The password for the database administrator account. This is an 8-41 character, alphanumeric string and must not include white space, forward slashes (/), or backslashes (\). The user name is root .
Multi-AZ Database (DBMultiAZ)	true	Set to false if Multi-AZ deployment isn't needed for the DB instances.

DNS and SSL Configuration:

Parameter label (name)	Default	Description
Domain Name (DomainName)	<i>Optional</i>	The domain name to use for the website. This must be an existing, publicly resolvable domain. If you don't specify a

Parameter label (name)	Default	Description
		zone ID for the Route 53 Hosted Zone ID parameter, you will need to have access to the email address defined in the domain's start of authority (SOA) record and accept the ACM validation email that is sent during the creation of the Quick Start. For more information, see the ACM documentation . If you leave this parameter blank, no SSL certificate will be generated. However, certificates will be installed if you use the SSL Certificate ARN parameter.
SSL Certificate ARN (CertificateArn)	<i>Optional</i>	The Amazon Resource Name (ARN) of the SSL certificate to use for the load balancer. If you leave this parameter blank but specify a domain name in the previous parameter, the certificate will be auto-generated. If you leave both parameters blank, no SSL certificate will be used.
Route 53 Hosted Zone ID (HostedZoneID)	<i>Optional</i>	The Amazon Route 53 hosted zone ID to use. If you leave this parameter blank, the Quick Start will not configure Route 53, and you must set up the DNS manually, as outlined in the ACM documentation . Use this parameter only if you have specified a domain name in the Domain Name parameter.

WordPress Webserver Configuration:

Parameter label (name)	Default	Description
Admin Password (WordpressAdmin Password)	<i>Requires input</i>	The password for the WordPress site administrator account. This is an 8-41 character, alphanumeric string. The user name is user .
Instance Size (WebServerInstanceType)	t2.small	The Amazon EC2 instance type for the WordPress instances.
Instance enhanced monitoring (WebServerInstance Monitoring)	Enabled	Set this parameter to Disabled if you want to turn off enhanced monitoring for WordPress instances. When enhanced monitoring is enabled, the Amazon EC2 console displays monitoring graphs with 1-minute resolution. For more information, see Enable or Disable Detailed Monitoring for Your Instances in the Amazon EC2 documentation.
Min Number of Instances (WebServerMinSize)	1	The minimum number of EC2 instances in the Auto Scaling group of WordPress instances.
Max Number of Instances (WebServerMaxSize)	12	The maximum number of EC2 instances in the Auto Scaling group of WordPress instances.
Desired Number of Instances (WebServerDesired Capacity)	2	The desired number of EC2 instances in the Auto Scaling group of WordPress instances.

Parameter label (name)	Default	Description
Autoscaling Notification Email (AutoScalingNotificationEmail)	<i>Requires input</i>	The email address to notify if there are any scaling operations.

ElastiCache Configuration:

Parameter label (name)	Default	Description
Enable ElastiCache (ElastiCacheEnable)	true	Amazon ElastiCache for Memcached provides an in-memory key-value store service for caching database queries and improving performance. Set this parameter to false if you don't want to enable ElastiCache for WordPress.
Enable ElastiCache Auto Minor Version Upgrade (ElastiCacheAutoMinorVersionUpgrade)	true	Determines whether the ElastiCache node will automatically be upgraded to a new, minor version of the Memcached engine supported by ElastiCache. Set this parameter to false to disable auto-upgrades.
ElastiCache Node Type (ElastiCacheNodeType)	cache.t2.micro	The node type that determines the compute and memory capacity of nodes in a cache cluster.
Number of ElastiCache Nodes (ElastiCacheNumberOfNodes)	2	The number of cache nodes that the cache cluster should have. You can set this parameter to a value between 1 and 20.

AWS Quick Start Configuration:

Parameter label (name)	Default	Description
Quick Start S3 Bucket Name (QSS3BucketName)	aws-quickstart	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.
Quick Start S3 Key Prefix (QSS3KeyPrefix)	quickstart-bitnami-wordpress/	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.

- **Option 2: Parameters for deploying WordPress High Availability by Bitnami into an existing VPC**

[View template](#)

Network Configuration:

Parameter label (name)	Default	Description
VPC ID (VPCID)	<i>Requires input</i>	The ID of your existing VPC (e.g., vpc-0343606e).
VPC CIDR (VPCCIDR)	<i>Requires input</i>	The CIDR block for the VPC.
Private Subnet-1 ID (PrivateSubnet1ID)	<i>Requires input</i>	The ID of the private subnet in Availability Zone 1 in your existing VPC (e.g., subnet-a0246dcd).
Private Subnet-2 ID (PrivateSubnet2ID)	<i>Requires input</i>	The ID of the private subnet in Availability Zone 2 in your existing VPC (e.g., subnet-b58c3d67).
Public Subnet-1 ID (PublicSubnet1ID)	<i>Requires input</i>	The ID of the public subnet in Availability Zone 1 in your existing VPC (e.g., subnet-b1f4a2cd).
Public Subnet-2 ID (PublicSubnet2ID)	<i>Requires input</i>	The ID of the public subnet in Availability Zone 2 in your existing VPC (e.g., subnet-9bc642ac).
Allowed CIDR for ALB Access (ALBAccessCIDR)	127.0.0.1/32	The CIDR IP range that is permitted external web access to the Application Load Balancer. We recommend that you set this value to a trusted IP range.
Bastion Security Group ID (BastionSecurityGroupID)	<i>Requires input</i>	The ID of the bastion security group in your existing VPC (e.g., sg-1d2c3b4a).

Aurora Database Configuration:

Parameter label (name)	Default	Description
Enable Auto Minor Version Upgrade (DBAutoMinorVersionUpgrade)	true	Determines whether the DB instance will automatically be upgraded to new Aurora or MySQL minor versions as they are supported by Amazon RDS. Set this parameter to false to disable auto-upgrades.
Backup Retention Period (DBBackupRetentionPeriod)	7	Number of days for which automatic DB snapshots are retained. You can set this parameter to a value between 1 and 35 days.
Preferred Backup Window (DBPreferredBackupWindow)	<i>Optional</i>	The preferred time period for automated backups, specified in the format HH:MM-HH:MM (daily start time-daily end time), using coordinated universal time (UTC). For example, 09:15-13:15 would set the start time to 9:15 AM UTC, and the end time to 1:15 PM UTC. Backups that are created during this

Parameter label (name)	Default	Description
		window are retained for the number of days specified by the Backup Retention Period parameter. For more information, see the Amazon RDS documentation .
Database Instance Size (DBInstanceClass)	db.t2.small	The instance size (compute and memory capacity) class for the Aurora DB instances.
Database Admin Password (DBMasterUserPassword)	<i>Requires input</i>	The password for the database administrator account. This is an 8-41 character, alphanumeric string and must not include white space, forward slashes (/), or backslashes (\). The user name is root .
Multi-AZ Database (DBMultiAZ)	true	Set to false if Multi-AZ deployment isn't needed for the DB instances.

DNS and SSL Configuration:

Parameter label (name)	Default	Description
Domain Name (DomainName)	<i>Optional</i>	The domain name to use for the website. This must be an existing, publicly resolvable domain. If you don't specify a zone ID for the Route 53 Hosted Zone ID parameter, you will need to have access to the email address defined in the domain's start of authority (SOA) record and accept the ACM validation email that is sent during the creation of the Quick Start. For more information, see the ACM documentation . If you leave this parameter blank, no SSL certificate will be generated. However, certificates will be installed if you use the SSL Certificate ARN parameter.
SSL Certificate ARN (CertificateArn)	<i>Optional</i>	The Amazon Resource Name (ARN) of the SSL certificate to use for the load balancer. If you leave this parameter blank but specify a domain name in the previous parameter, the certificate will be auto-generated. If you leave both parameters blank, no SSL certificate will be used.
Route 53 Hosted Zone ID (HostedZoneID)	<i>Optional</i>	The Amazon Route 53 hosted zone ID to use. If you leave this parameter blank, the Quick Start will not configure Route 53, and you must set up the DNS manually, as outlined in the ACM documentation . Use this parameter only if you have specified a domain name in the Domain Name parameter.

WordPress Webserver Configuration:

Parameter label (name)	Default	Description
Admin Password (WordpressAdmin Password)	<i>Requires input</i>	The password for the WordPress site administrator account. This is an 8-41 character, alphanumeric string. The user name is user .

Parameter label (name)	Default	Description
Instance Size (WebServerInstanceType)	t2.small	The Amazon EC2 instance type for the WordPress instances.
Instance enhanced monitoring (WebServerInstanceMonitoring)	Enabled	Set this parameter to Disabled if you want to turn off enhanced monitoring for WordPress instances. When enhanced monitoring is enabled, the Amazon EC2 console displays monitoring graphs with 1-minute resolution. For more information, see Enable or Disable Detailed Monitoring for Your Instances in the Amazon EC2 documentation.
SSH Keypair Name (KeyPairName)	<i>Requires input</i>	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.
Min Number of Instances (WebServerMinSize)	1	The minimum number of EC2 instances in the Auto Scaling group of WordPress instances.
Max Number of Instances (WebServerMaxSize)	12	The maximum number of EC2 instances in the Auto Scaling group of WordPress instances.
Desired Number of Instances (WebServerDesiredCapacity)	2	The desired number of EC2 instances in the Auto Scaling group of WordPress instances.
Autoscaling Notification Email (AutoScalingNotificationEmail)	<i>Requires input</i>	The email address to notify if there are any scaling operations.

ElastiCache Configuration:

Parameter label (name)	Default	Description
Enable ElastiCache (ElastiCacheEnable)	true	Amazon ElastiCache for Memcached provides an in-memory key-value store service for caching database queries and improving performance. Set this parameter to false if don't want to enable ElastiCache for WordPress.
Enable ElastiCache Auto Minor Version Upgrade (ElastiCacheAutoMinorVersionUpgrade)	true	Determines whether the ElastiCache node will automatically be upgraded to a new, minor version of the Memcached engine supported by ElastiCache. Set this parameter to false to disable auto-upgrades.
ElastiCache Node Type (ElastiCacheNodeType)	cache.t2.micro	The node type that determines the compute and memory capacity of nodes in a cache cluster.

Parameter label (name)	Default	Description
Number of ElastiCache Nodes (ElastiCacheNumberOfNodes)	2	The number of cache nodes that the cache cluster should have. You can set this parameter to a value between 1 and 20.

AWS Quick Start Configuration:

Parameter label (name)	Default	Description
Quick Start S3 Bucket Name (QSS3BucketName)	aws-quickstart	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.
Quick Start S3 Key Prefix (QSS3KeyPrefix)	quickstart-bitnami-wordpress/	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.

- 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**.

CloudFormation > Stacks > Create Stack

Create stack

Select Template
Specify Details
Options
Review

Options

Tags

You can specify tags (key-value pairs) for resources in your stack. You can add up to 50 unique key-value pairs for each stack. [Learn more.](#)

	Key (127 characters maximum)	Value (255 characters maximum)	
1	<input type="text"/>	<input type="text"/>	<input type="button" value="+"/>

Permissions

You can choose an IAM role that CloudFormation uses to create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses the permissions defined in your account. [Learn more.](#)

IAM Role

Enter role arn

Rollback Triggers

Rollback triggers enable you to have AWS CloudFormation monitor the state of your application during stack creation and updating, and to rollback that operation if the application breaches the threshold of any of the alarms you've specified. [Learn more](#)

Monitoring Time Minutes
Minimum value of 0. Maximum value of 180.

Figure 5: Setting options

7. On the **Review** page, review and confirm the template settings. Under **Capabilities**, select the two check boxes to acknowledge that the template will create IAM resources and that it might require the capability to auto-expand macros.

Capabilities

i The following resource(s) require capabilities: [AWS::CloudFormation::Stack]

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#).

For this template, AWS CloudFormation might require an unrecognized capability: CAPABILITY_AUTO_EXPAND. Check the capabilities of these resources.

I acknowledge that AWS CloudFormation might create IAM resources with custom names.

I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY_AUTO_EXPAND

[Quick Create Stack](#) (Create stacks similar to this one, with most details auto-populated)

Cancel Previous Create

Figure 6: Accepting the creation of IAM resources and auto-expand

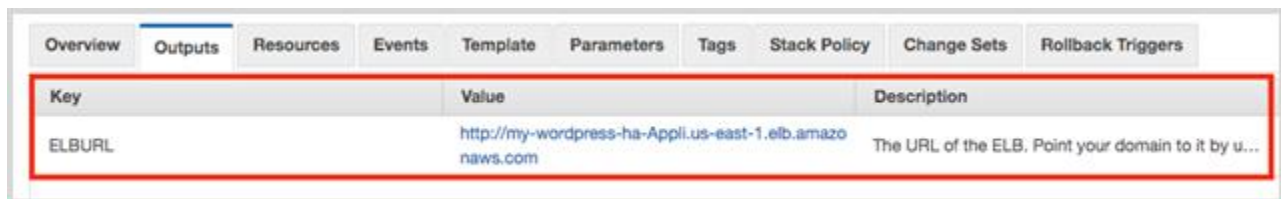
8. Choose **Create** to deploy the stack.
9. Monitor the status of the stack. When the status is **CREATE_COMPLETE**, the WordPress High Availability for Bitnami cluster is ready.
10. Use the URLs displayed in the **Outputs** tab for the stack to view the resources that were created.

Step 3. Test the Deployment

When you see that the deployment has completed successfully, you can test it by either accessing the WordPress user interface and logging in to the dashboard, or by connecting to the cluster through SSH.

Option 1. Access the WordPress user interface

1. Open the [AWS CloudFormation console](#), and select the Bitnami stack.
2. In the stack details pane, choose the **Outputs** tab.
3. Choose the link in the **Value** column for **ELBURL**, as shown in Figure 7.



Key	Value	Description
ELBURL	http://my-wordpress-ha-Appil.us-east-1.elb.amazonsaws.com	The URL of the ELB. Point your domain to it by u...

Figure 7: Link to WordPress in Outputs tab

This will display your WordPress blog's home page with a sample post, as shown in Figure 8.

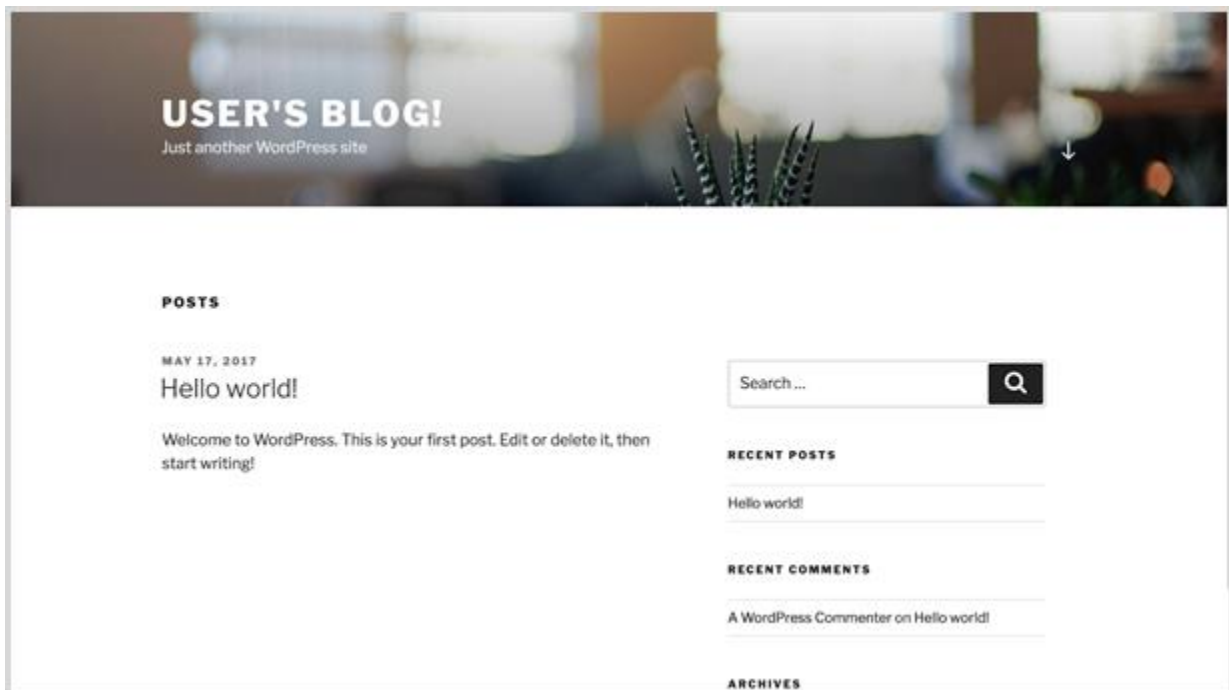


Figure 8: WordPress blog home page

To log in to the WordPress dashboard to manage your blog posts, follow these steps:

1. Open the WordPress dashboard by adding `/wp-admin` to the URL you used to access WordPress.
2. Log in with the user name **user** and the password you specified for the **Admin Password** (WordPressAdminPassword) parameter during the deployment process.

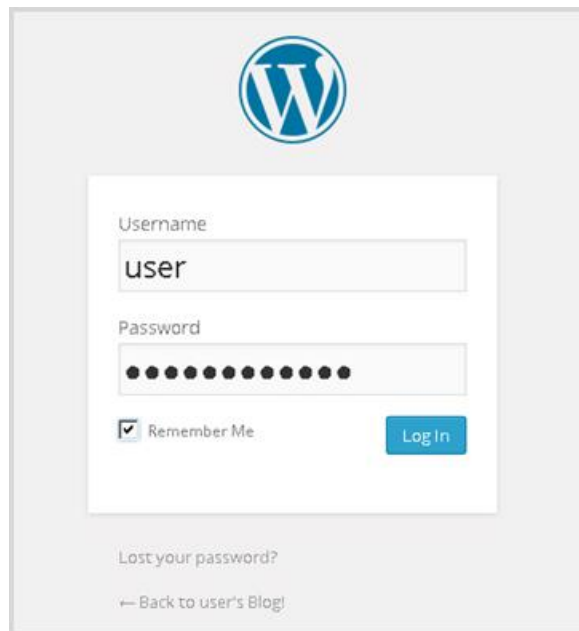


Figure 9: Entering WordPress administrator credentials

This will display the WordPress dashboard, which allows you to manage posts, pages, and comments; customize your blog with themes and plugins; import and export content; manage navigation menus; add or delete new user accounts; and much more.

Option 2. Connect to the cluster through SSH

To connect to the WordPress cluster through SSH, you must first connect to the bastion host instance, and then connect through SSH to the WordPress instance. To forward the SSH key, see [Securely Connect to Linux Instances Running in a Private Amazon VPC](#) in the AWS Security Blog.

1. Open the [AWS CloudFormation console](#), and select the nested stack called **Bastion Stack**.
2. From the **Outputs** tab, copy the IP address for the **EIP1** key.

Key	Value	Description
EIP1	██████████	Elastic IP 1 for Bastion

Figure 10: IP address for EIP1

- Open a new terminal window and run the following command, specifying the IP address from the previous step:

```
ssh ec2-user@EIP1_IP_ADDRESS
```

This will connect you to the bastion host instance. Now you can get the IP address for the Bitnami instance and connect to it via SSH.

- Open the [Amazon EC2 console](#), choose **Instances** from the navigation pane, and select the Bitnami instance. In the **Description** tab, copy the private IP address for the instance.

Instance: [redacted] (wp-ha-2-1 [redacted] - Web Server) Private IP: [redacted]	
<div style="display: flex; justify-content: space-between;"> Description Status Checks Monitoring Tags </div>	
Instance ID	i-011d09d5a3411e601
Instance state	running
Instance type	t2.micro
Elastic IPs	
Availability zone	us-east-1b
Public DNS (IPv4)	-
IPv4 Public IP	-
IPv6 IPs	-
Private DNS	ip-1[redacted].internal
Private IPs	██████████

Figure 11: Private IP address for Bitnami instance

- In the terminal window, run the following command to connect to the Bitnami instance, specifying the IP address from the previous step:

```
ssh bitnami@IP_ADDRESS
```

- From the Bitnami instance, run the following command to check if the services are running. You can also check how the shared file system (Amazon EFS) is mounted or reveal monitoring statistics with Gonit:

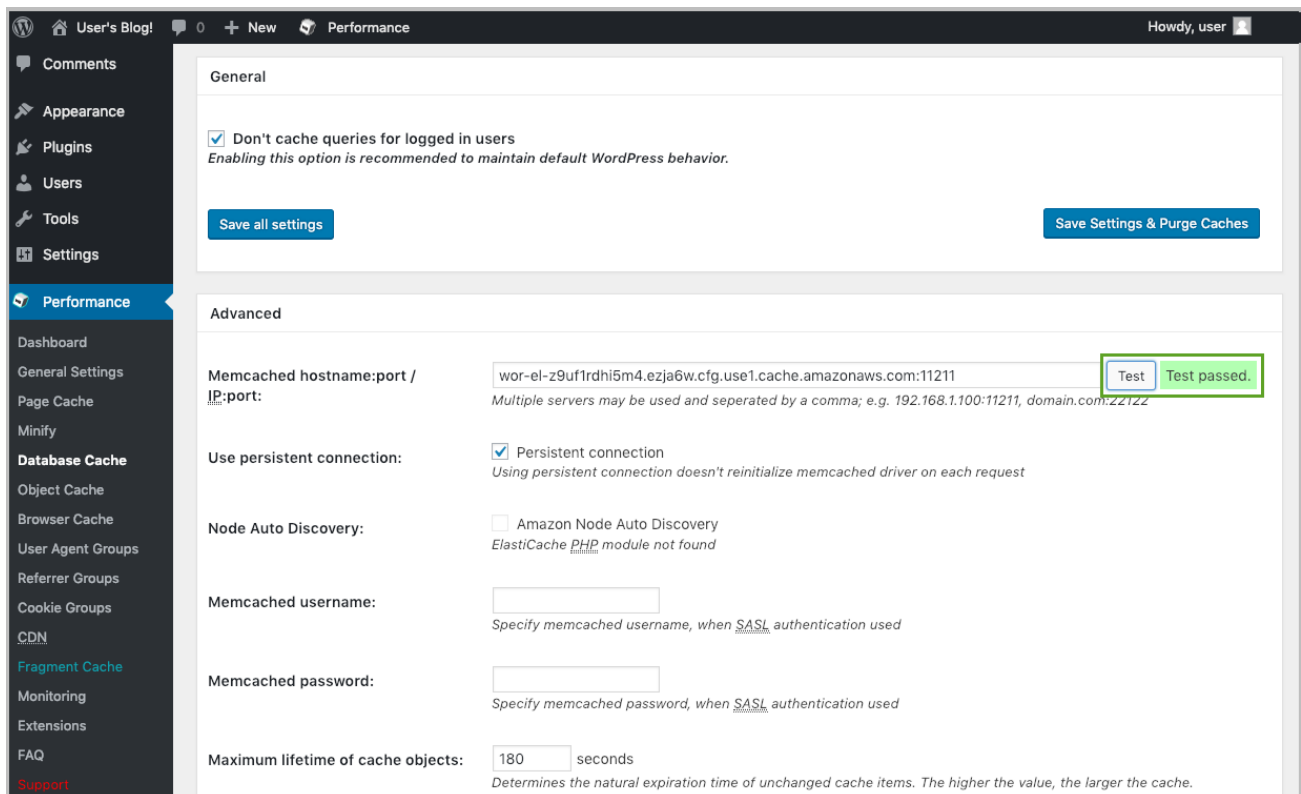
```
sudo service bitnami status
df -h
sudo gonit status
```

Step 4. (Optional) Test the W3 Total Cache Plugin

If you enable ElastiCache during deployment by keeping the default setting for the **Enable ElastiCache** (ElastiCacheEnable) parameter, the Quick Start will create a Memcached cluster. WordPress instances use the W3 Total Cache plugin to configure the WordPress cache. This Quick Start activates the plugin and configures it to use the ElastiCache deployment as its database backend.

To check that the plugin is working correctly, follow these steps:

1. Log in to the WordPress dashboard.
2. Choose **Performance, Database Cache**. Use the **Test** button to confirm that the plugin is working correctly.



The screenshot shows the WordPress Performance settings page. The left sidebar is expanded to 'Performance', and the 'Database Cache' section is active. The 'Memcached hostname:port / IP:port' field contains the value 'wor-el-29uf1rdhi5m4.ezja6w.cfg.use1.cache.amazonaws.com:11211'. A 'Test' button is highlighted with a green box, and a 'Test passed.' message is visible next to it. Other settings include 'Use persistent connection' (checked), 'Node Auto Discovery' (unchecked), 'Memcached username' (empty), 'Memcached password' (empty), and 'Maximum lifetime of cache objects' (180 seconds).

Figure 12: W3 Total Cache database cache settings

Step 5. (Optional) Configure OPcache

This Quick Start enables OPcache, which is PHP's byte code cache, with the following configuration by default:

Function	Meaning	Value set by the Quick Start
<code>opcache.realpath_cache_size</code>	The size of the realpath cache to be used by PHP.	4,096 KB
<code>opcache.max_accelerated_files</code>	The maximum number of PHP files that OPcache can hold at once.	4,000 files
<code>opcache.memory_consumption</code>	The size of the cached data that memory can hold. Increasing this value increases the performance.	192 MB
<code>opcache.file_cache</code>	The local directory to hold cache data, in case memory size turns out to be a limiting factor <code>opcache.memory_consumption</code>.	<code>/opt/bitnami/php/tmp/file_cache</code>
<code>opcache.validate_timestamps</code>	When enabled, OPcache will automatically check file timestamps for any changes to cached files.	1 (enabled)

If you want to edit the configuration, follow these steps:

1. [Connect to your WordPress instance by using SSH.](#)
2. Edit the file `/opt/bitnami/php/lib/php.ini` to change the settings of these and other functions. For a full list, see the [PHP documentation](#).
3. Restart the Apache service with the following command:

```
sudo service bitnami restart apache
```

Reconfiguration notes:

- Before you reconfigure the `opcache.max_accelerated_files` function, you should find the current number of PHP files:

```
find /opt/bitnami/apps/wordpress -type f -print | grep -c php
```

- For production environments, consider setting `opcache.validate_timestamps` to 0 (zero). This disables the checking for the timestamp of the cached files and improves performance. However, if you disable this check, whenever you add or modify PHP files

you will need to restart the Apache service in order for the changes in those files to take effect.

To visit the OPcache dashboard, follow these steps:

1. Open the OPcache dashboard by adding `/opcache-dashboard.php` to the URL you used to access WordPress.
2. Log in with the user name **user** and the password you specified for the **Admin Password** (WordPressAdminPassword) parameter during the deployment process.

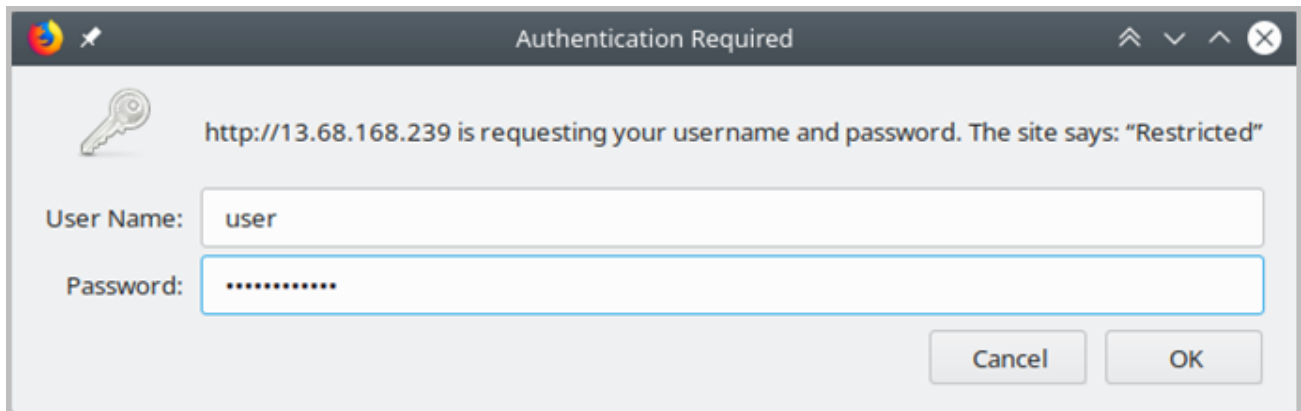


Figure 13: Logging in to the OPcache dashboard

This will display the OPcache dashboard, which shows you the status of the cache, as illustrated in Figure 14.

PHP 7.1.23 with OpCache 7.1.23

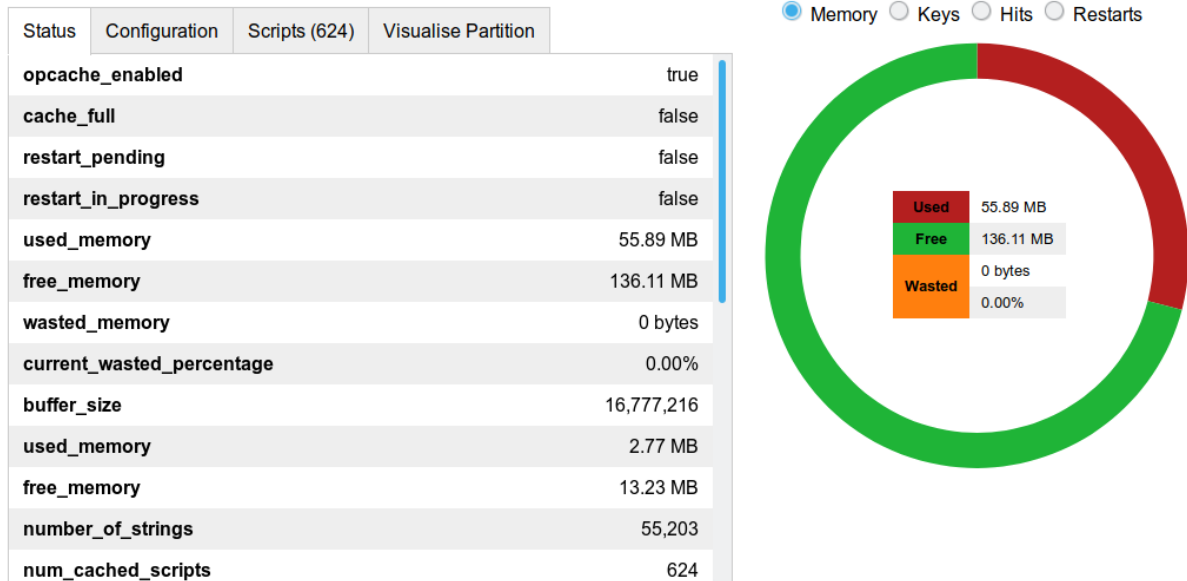


Figure 14: OpCache dashboard

Best Practices Using WordPress High Availability by Bitnami on AWS

To keep your WordPress deployment up-to-date and secure, see the following documentation on the Bitnami website:

- [How to enforce WordPress security](#)
- [How to connect to the Amazon RDS database](#)
- [How to reset the Amazon RDS database password](#)

WordPress Updates

There are three ways to keep your WordPress deployment updated:

- Update WordPress components manually
- Get minor WordPress version updates from the dashboard
- Get major release updates from Bitnami

Updating WordPress Components Manually

You can update the version of WordPress components at any time by connecting to the cluster through SSH, as explained [earlier in this guide](#), and running the following command:

```
sudo apt-get update
```

Updating WordPress from the Dashboard

You can easily update to the next minor version of WordPress from the WordPress dashboard:

1. Log in to WordPress using the administrator account.
2. Choose **Dashboard, Updates**.
3. Choose the **Update Now** button, which will be displayed if an update is available.

You can use this method to get automatic security updates as well as bug fixes and minor feature updates.

Getting Major Release Updates from Bitnami

When a major version of WordPress becomes available, you will get a notification in the WordPress dashboard. For each major update, Bitnami provides an updated and secure release that includes new versions of operating system packages, libraries, components, and dependencies such as Apache or PHP. Before updating the software, we recommend that you create a backup of your site, including the database, by downloading and installing the [BackUpWordPress plugin](#).

After you back up your site, you can update your version of WordPress by redeploying the AWS CloudFormation template:

**Relaunch Quick Start
to get major updates**

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, 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 Quick Start leverages 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.

Operating System Security

WordPress High Availability by Bitnami includes these security features:

- **SSH key access:** 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.
- **Authentication:** WordPress High Availability by Bitnami configures several security enhancements by default, such as an administrator and SSH user name set. During deployment, you will be prompted to enter administrator and database passwords. Make sure that you enter strong and highly secure passwords.
- **Network and access ports:** For security reasons, WordPress High Availability by Bitnami is not reachable from external networks. All ports are closed by default except for port 80. If you need to connect to the cluster remotely from a different network, you can peer both virtual networks by using network peering or a secure channel such as a VPN or an SSH tunnel.
- **Updates:** If serious security issues are discovered, Bitnami provides new versions of the application, often within hours of a fix becoming available. For more information, see the [WordPress Updates](#) section of this guide.

Other Security Considerations

Follow these additional guidelines to keep your deployment secure:

- **Add a second authentication layer** to your Apache configuration by following [these instructions](#) on the Bitnami website.
- **Keep the pingback functionality disabled.** WordPress implements an interface that uses the XML-RPC protocol for features such as link notification (pingback) and remote publishing from web clients and smartphone apps. The pingback feature of XML-RPC is known to be susceptible to brute force amplification and distributed denial

of service (DDOS) attacks, so Bitnami has disabled it by default. For more information, see the [Bitnami documentation](#).

- **Create backups** periodically.
- **Keep your deployment updated.** We recommend that you [update your WordPress deployment](#) periodically to make sure that you have the latest versions of components, dependencies, and security patches.

Before updating the deployment, you should create a backup of your site. Download and install the [BackUpWordPress plugin](#) to create a copy of your entire site, including the database. For more information, see the [WordPress Updates](#) section earlier in this guide.

For more information about WordPress security, see the [WordPress documentation](#) on the WordPress.org website.

FAQ

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.

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.

For additional information, see [Troubleshooting AWS CloudFormation](#) on the AWS website.

Q. I encountered a size limitation error when I deployed the AWS CloudFormation templates.

A. We recommend that you launch the Quick Start templates from the links in this guide or from another S3 bucket. If you deploy the templates from a local copy on your computer or from a non-S3 location, you might encounter template size limitations when you create the stack. For more information about AWS CloudFormation limits, see the [AWS documentation](#).

Bitnami Support for AWS

Bitnami provides specific support for AWS users. If you have specific questions about the WordPress High Availability by Bitnami software, see the [Bitnami support website](#). You can type a question or topic to find related content across all Bitnami sites. If you need further assistance, visit the [Bitnami Community Forums](#).

GitHub Repository

You can visit our [GitHub repository](#) to download the templates and scripts for this Quick Start, to post your comments, and to share your customizations with others.

Additional Resources

AWS services

- Amazon EBS
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html>
- Amazon EC2
<https://docs.aws.amazon.com/ec2/>
- Amazon VPC
<https://docs.aws.amazon.com/vpc/>
- AWS CloudFormation
<https://docs.aws.amazon.com/cloudformation/>
- Amazon ElastiCache
<https://docs.aws.amazon.com/elasticache/>

WordPress documentation

- Getting Started with WordPress
https://codex.wordpress.org/Getting_Started_with_WordPress
- Bitnami WordPress documentation for AWS
<https://docs.bitnami.com/aws-templates/apps/wordpress/>

Quick Start reference deployments

- AWS Quick Start home page
<https://aws.amazon.com/quickstart/>

Document Revisions

Date	Change	In sections
January 2019	Added ElastiCache, W3 Total Cache plugin, and OPcache support	Architecture , Step 2 (new parameters), Step 4 , Step 5
September 2018	Initial publication	—

© 2020, Amazon Web Services Inc., or its affiliates, and Bitnami. 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.