

# Jira Software and Jira Service Desk Data Center on the AWS Cloud

## Quick Start Reference Deployment

October 2016

([last update](#): April 2020)

Chris Szmajda, Felix Haehnel – Atlassian

Shiva Narayanaswamy, Tony Vattathil – Amazon Web Services

### Contents

Overview.....	2
Costs and licenses.....	2
Architecture.....	4
Amazon Aurora PostgreSQL.....	5
Auto Scaling groups in this Quick Start.....	6
Prerequisites .....	7
Specialized knowledge.....	7
Technical requirements.....	7
Deployment options.....	7
Deployment steps.....	8
Step 1. Prepare an AWS account .....	8
Step 2. Launch the Quick Start .....	8
Step 3. Review the parameters and create your stack .....	10
Step 4. Configure Jira.....	23
Step 5. Add nodes to the cluster.....	27
Troubleshooting.....	28
Additional resources .....	29

Git repository .....	30
Document revisions .....	30

This Quick Start deployment guide was created by Atlassian 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.

## Overview

This Quick Start reference deployment guide provides step-by-step instructions for deploying Atlassian Jira Data Center applications on the AWS Cloud.

The Quick Start deploys two Jira solutions from Atlassian:

- **Jira Software Data Center** is a software development tool that is used by agile teams. It helps development teams create and estimate projects, build a sprint backlog, identify team commitments and velocity, visualize team activity, and report on progress. For more information, see the [Jira Software documentation](#) on the Atlassian website.
- **Jira Service Desk Data Center** provides IT service management software with high availability and performance at scale on AWS. Jira Service Desk is easy to use, simple to set up, and has everything you need for IT support and customer service. Teams can link Jira Service Desk tickets to Jira Software issues and keep development and IT connected for fast issue resolution. Jira Service Desk Data Center is a self-hosted solution that provides high availability and disaster recovery, so no service request goes unresolved. For more information, see the [Jira Service Desk documentation](#) on the Atlassian website.

Please know that we may share who uses AWS Quick Starts with the AWS partner that collaborated with AWS on the content of the Quick Start.

## 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 templates 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](#).

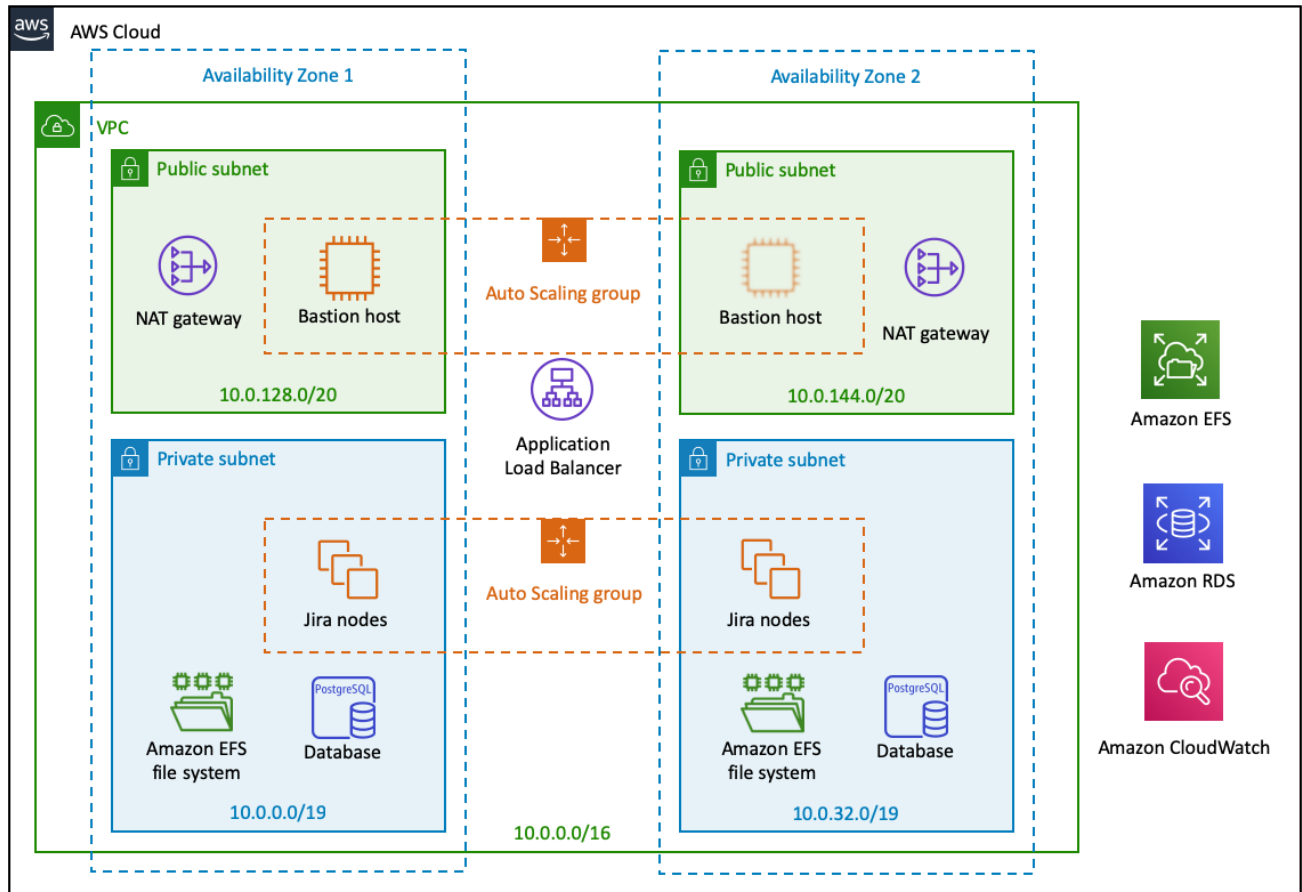
This Quick Start requires a license for either Jira Software Data Center or Jira Service Desk Data Center. For information about obtaining a license, see the [Atlassian pricing page](#).

**Note:** This Quick Start can't be used with Jira Software or Jira Service Desk Server licenses.

If you want to evaluate either Jira Software Data Center or Jira Service Desk Data Center, you can also request an evaluation license after you deploy the Quick Start, during application configuration.

## Architecture

Deploying this Quick Start with **default parameters** builds the following Jira Data Center environment in the AWS Cloud.



**Figure 1: Jira architecture on AWS**

This Quick Start includes two AWS CloudFormation templates: the first one builds out the Atlassian Standard Infrastructure (ASI), which is a virtual private cloud (VPC) that contains the components required by all Atlassian Data Center applications, and then provisions Jira software into this ASI. The second template provisions Jira in an existing ASI.

The resulting infrastructure from this Quick Start contains the following:

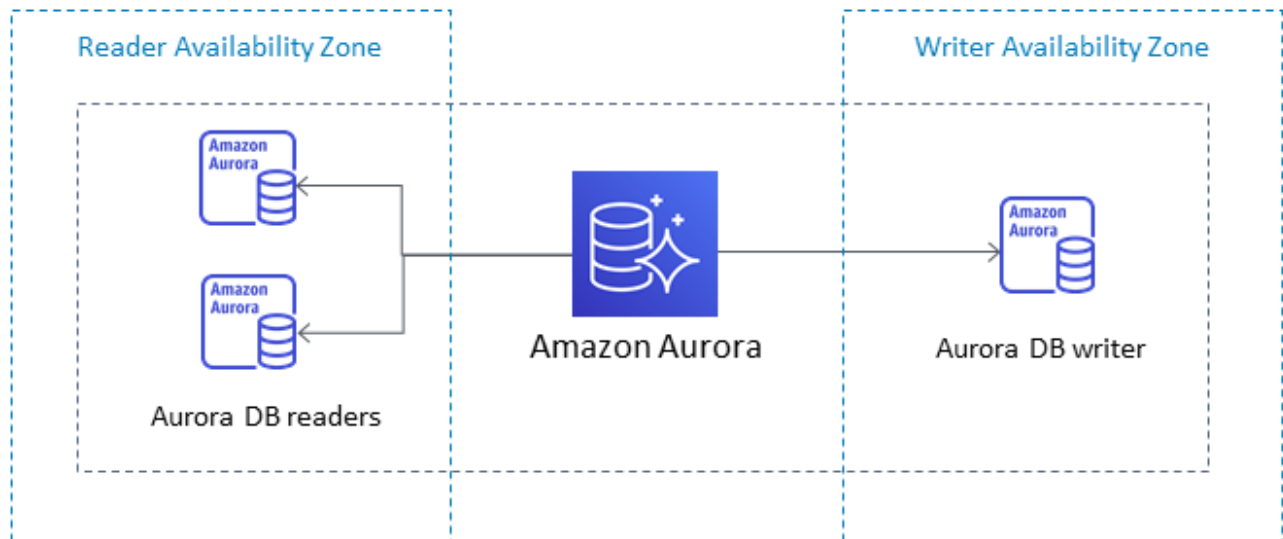
- **Amazon RDS PostgreSQL:** Jira applications require a supported external database. Amazon Relational Database Service (Amazon RDS) PostgreSQL in a Multi-AZ configuration allows failover in the event the master node fails. Alternatively, you can use a more resilient [Amazon Aurora PostgreSQL](#) clustered database.

- **Amazon Elastic File System (Amazon EFS):** Jira applications use a shared file system to store artifacts in a common location that is accessible to multiple Jira nodes. The Quick Start architecture implements the shared file system using the highly available Amazon EFS service.
- **Jira Auto Scaling group:** The Jira applications are installed on Amazon Elastic Compute Cloud (Amazon EC2) instances in an Auto Scaling group. The instances are based on Amazon Linux and use an Atlassian-provided Amazon Machine Image (AMI). The scaling metric is CPU utilization.
- **Bastion host:** This host enables secure access to Jira without exposing it to the internet. (For more information, see [Bastion Hosts](#).)
- **An Application Load Balancer:** This works both as a load balancer and a Secure Sockets Layer (SSL) termination reverse proxy.
- **Amazon CloudWatch:** This provides basic monitoring for all application and database nodes in your deployment. By default, CloudWatch will also collect and store logs from each monitored node. This Quick Start configures a CloudWatch dashboard to help you read and analyze collected data. You can configure this dashboard with additional metrics and monitoring after deployment.

## Amazon Aurora PostgreSQL

You can also use this Quick Start to deploy Jira Data Center with an Aurora clustered database that's compatible with PostgreSQL. The cluster configuration is illustrated in Figure 2. It features a primary database writer that replicates to two database readers in a different Availability Zone. If the writer fails, Aurora promotes one of the readers to take its place with no downtime. For more information, see [Amazon Aurora Features: PostgreSQL-Compatible Edition](#).

The Aurora configuration with a single database writer and two readers is designed to provide high availability without degrading performance.



**Figure 2: Aurora reader/writer configuration**

Amazon Aurora is supported on Jira Software Data Center 8.4, Jira Service Desk 4.4, and all later versions of both.

### Auto Scaling groups in this Quick Start

This Quick Start uses [Auto Scaling groups](#), but only to statically control the number of its nodes. We don't recommend that you use Auto Scaling to dynamically scale the size of your cluster. Adding an application node to the cluster usually takes more than 20 minutes, which isn't fast enough to address sudden load spikes.

If you can identify any periods of high and low load, you can schedule the application node cluster to scale accordingly. See [Scheduled Scaling for Amazon EC2 Auto Scaling](#) for more information.

To study trends in your organization's load, you'll need to monitor the performance of your deployment. Refer to [Jira Data Center sample deployment and monitoring strategy](#) for tips on how to do so.

## 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 the [Getting Started Resource Center](#).)

- [Amazon Virtual Private Cloud \(Amazon VPC\)](#)
- [Amazon EC2](#)
- [Amazon EFS](#)
- [PostgreSQL on Amazon RDS](#)

### Technical requirements

This Quick Start requires an Atlassian account and a Data Center license to use the Jira Data Center applications you're planning to deploy on AWS, as discussed earlier in the [Costs and Licenses](#) section.

## Deployment options

This Quick Start provides two deployment options:

- **Deploy Jira into a new ASI (end-to-end deployment).** Choose this option if you're a new user. This option builds the Atlassian Standard Infrastructure (ASI), which is a VPC that consists of the subnets, NAT gateways, security groups, bastion hosts, and other infrastructure components required by all Atlassian applications, and then deploys Jira into this new VPC.
- **Deploy Jira into an existing ASI.** Choose this option if you've already deployed the ASI separately by using the [ASI Quick Start](#), or by deploying another Atlassian product ([Bitbucket](#) or [Confluence Data Center](#)). This option provisions Jira in your existing ASI.

**Important:** You can provision Jira into an ASI that has an existing Atlassian application (for example, Bitbucket). If you are deploying and integrating multiple Atlassian applications, we recommend that you provision them all in the same ASI.

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

## Deployment steps

### Step 1. Prepare an AWS account

1. If you don't already have an AWS account, create one at <http://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 your Jira application on AWS.
3. Create a [key pair](#) in your preferred region.
4. If necessary, [request a service quota increase](#) for the Amazon EC2 **t3.medium** instance type. You might need to do this if you already have an existing deployment that uses this instance type, and you think you might exceed the [default quota](#) with this reference deployment.

### 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.

There are two methods of deploying Jira into your AWS account. The fastest way is by [launching the Quick Start directly from our Amazon S3 bucket](#). However, when you do, any updates we make to the Quick Start templates will propagate directly to your deployment. These updates sometimes involve adding or removing parameters from the templates. This could introduce unexpected (and possibly breaking) changes to your deployment.

As such, for production environments, we recommend a second method, where you [copy the Quick Start templates into your own S3 bucket](#). Then, launch them directly from there. Launching from your own templates lets you control when to apply our latest changes to your environment.



*Method 1: Launch the Quick Start from its original Amazon S3 bucket*

1. Use one of the following options to deploy Jira into your AWS account. For help choosing an option, see [deployment options](#) earlier in this guide.

Launch Quick Start  
(with new ASI)

Launch Quick Start  
(for existing ASI)

Deploying Jira into a new ASI takes about 50 minutes. Deploying Jira into an existing ASI takes about 30 minutes.

2. Check the AWS 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 your AWS environment will be built. The template is launched in the US West (Oregon) Region by default.

**Note:** This deployment includes Amazon EFS, which isn't currently supported in all AWS Regions. For a current list of supported regions, see the [endpoints and quotas webpage](#).

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

At this point, you can go directly to [Step 3: Review the parameters and create your stack](#) to continue.

*Method 2 (recommended): Launch the Quick Start from your own S3 bucket*

1. Clone a local copy of the [Quick Start templates](#) (including all of its submodules). From the command line, run:

```
git clone --recurse-submodules https://github.com/aws-quickstart/quickstart-atlassian-jira.git
```

2. **(Optional)** The Quick Start templates repository uses the directory structure required by the Quick Start interface. If needed (for example, to minimize storage costs), you can remove all other files except the following:

```
quickstart-atlassian-jira
├── submodules
│   ├── quickstart-atlassian-services
│   │   └── templates
│   └── quickstart-vpc-for-atlassian-services.yaml
```

```
└── templates
    ├── quickstart-jira-dc-with-vpc.template.yaml
    └── quickstart-jira-dc.template.yaml
```

3. Install and set up the AWS Command Line Interface (CLI). This tool enables you to create an S3 bucket and upload content to it.
4. Create an S3 bucket in your region:

```
aws s3 mb s3://<bucket-name> --region <AWS_REGION>
```

**Note:** This deployment includes Amazon EFS, which isn't currently supported in all AWS Regions. For a current list of supported regions, see the [endpoints and quotas webpage](#).

5. Choose which Quick Start template you'll be using and open it:
  - a. **quickstart-jira-dc-with-vpc.template.yaml**: use this for deploying Jira into a new ASI (end-to-end deployment).
  - b. **quickstart-jira-dc.template.yaml**: use this for deploying Jira into an existing ASI.
6. In the template you've chosen, the **QSS3BucketName** default value is set to **aws-quickstart**. Replace this default with the name of the bucket you created earlier.
7. Go into the parent directory of your local clone of the Quick Start templates. From there, upload all the files in your local clone to your S3 bucket:

```
aws s3 cp quickstart-atlassian-jira s3://<bucket-name> --recursive --acl public-read
```

Once you've uploaded everything, you're ready to deploy your production stack from your S3 bucket. Go to **CloudFormation** → **Create Stack**. When specifying a template, paste in the Object URL of the Quick Start template you'll be using.

You can now continue to [Step 3: Review the parameters and create your stack](#).

### Step 3. Review the parameters and create your stack

At this point, you should have launched the Quick Start. Whether you launched it from its [original S3 bucket](#) or [from your own](#), the next steps are the same:

1. On the **Specify Details** page, review the parameters for the template. Enter values for the parameters that require your input. For all other parameters, you can customize the default settings provided by the template.

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

- [Parameters for deploying Jira with a new ASI](#)
- [Parameters for deploying Jira into an existing ASI](#)

When you finish reviewing and customizing the parameters, choose **Next**.

- **Parameters for deploying Jira with a new ASI**

[View template](#)

*Jira setup:*

Parameter label (name)	Default	Description
<b>Jira Product *</b> (JiraProduct)	Software	The Jira product to install. You can choose <b>Core</b> (for Jira Core), <b>Software</b> (for Jira Software), or <b>ServiceDesk</b> (for Jira Service Desk).
<b>Version *</b> (JiraVersion)	7.13.3	The version of Jira to install. For a list of versions, see the release notes for <a href="#">Jira Software</a> and <a href="#">Jira Service Desk</a> on the Atlassian website.

*Cluster nodes:*

Parameter label (name)	Default	Description
<b>Enable CloudWatch Integration</b> (CloudWatchIntegration)	Metrics and Logs	Controls the extent to which Amazon CloudWatch should monitor your deployment. <b>Metrics and Logs</b> monitors node CPU, disk, and network activity, while also performing centralized logging. If the costs incurred by Amazon CloudWatch are an issue, use <b>Metrics Only</b> (to disable logging) or <b>Off</b> (to disable it completely).
<b>Cluster node instance type</b> (ClusterNodeInstanceType)	c5.xlarge	The EC2 instance type for the Jira application cluster nodes.
<b>Maximum number of cluster nodes</b> (ClusterNodeMax)	1	The maximum number of Jira nodes in the cluster. When you first launch the Quick Start, leave the default value of 1 node unchanged. In <a href="#">step 3</a> , after configuring Jira for multinode clustering, you can change this parameter setting to the desired number of cluster nodes.

Parameter label (name)	Default	Description
<b>Minimum number of cluster nodes</b> (ClusterNodeMin)	1	The minimum number of Jira nodes in the cluster. Leave this parameter at its default setting for a new deployment. You can update it after deployment.
<b>Cluster node instance volume size</b> (ClusterNodeVolumeSize)	50	The size of the cluster node root volume, in GB. You should base this size on the number of application indexes multiplied by 4.
<b>Deployment Automation Git Repository URL</b> (DeploymentAutomationRepository)	<a href="https://bitbucket.org/atlassian/dc-deployments-automation.git">https://bitbucket.org/atlassian/dc-deployments-automation.git</a>	The deployment automation repository to use for per-node initialization. Leave this as default unless you have customizations.
<b>Deployment Automation Branch</b> (DeploymentAutomationBranch)	master	The deployment automation repository branch to pull from.
<b>The Ansible playbook to invoke to initialize the instance</b> (DeploymentAutomationPlaybook)	aws_jira_dc_node.yml	The Ansible playbook to invoke to initialise the Jira node on first start.
<b>SSH keyname to use with the repository</b> (DeploymentAutomationKeyName)	<i>Optional</i>	Named Key Pair name to use with this repository. The key should be imported into the SSM parameter store.
<b>Custom command-line parameters for Ansible</b> (DeploymentAutomationCustomParams)	<i>Optional</i>	Additional command-line options for the `ansible-playbook` command. See <a href="https://bitbucket.org/atlassian/dc-deployments-automation/src/master/README.md">https://bitbucket.org/atlassian/dc-deployments-automation/src/master/README.md</a> for more information about overriding parameters.

*Database:*

Parameter label (name)	Default	Description
<b>The database engine to deploy with</b> (DBEngine)	PostgreSQL	Database Engine to use for the application. Choose either PostgreSQL (for Amazon RDS) or Amazon Aurora PostgreSQL.
<b>Database instance class</b> (DBInstanceClass)	db.m5.large	The EC2 instance type for the Amazon RDS database.
<b>RDS Provisioned IOPS</b> (DBIops)	1000	The IOPS for database storage. This value is used only when the <b>Database storage type</b> parameter is set to

Parameter label (name)	Default	Description
		<b>Provisioned IOPS.</b> Allowed range is 1,000-30,000. The ratio of IOPS to allocated storage must be between 3 and 10.
<b>Master (admin) password *</b> (DBMasterUserPassword)	<i>Requires input</i>	The password for the master (postgres) account. Must be at least 8 characters and include 1 uppercase, 1 lowercase, 1 number, and 1 of the following symbols: ! # \$ % ' * : [ = , ] - _ @ + % &.
<b>Enable RDS Multi-AZ deployment</b> (DBMultiAZ)	True	If <b>True</b> , your database deploys with high availability. When <b>DBEngine</b> equals <b>PostgreSQL</b> , a multi-AZ RDS instance is provisioned. When <b>DBEngine</b> equals <b>Amazon Aurora PostgreSQL</b> , a multi-node Amazon Aurora cluster is provisioned. If high availability isn't a concern, set this parameter to <b>False</b> .
<b>Application user database password</b> (DBPassword)	<i>Requires input</i>	Database password used by Jira. Must be at least 8 characters and include 1 uppercase, 1 lowercase, 1 number, and 1 of the following symbols: ! # \$ % ' * : [ = , ] - _ @ + % &.
<b>Database storage</b> (DBStorage)	200	The storage size, in GB, to allocate to the database. If you choose <b>Provisioned IOPS</b> for the <b>Database storage type</b> parameter, storage should be between 100 and 6144 GB.
<b>Database encryption</b> (DBStorageEncrypted)	False	Set this parameter to <b>True</b> if you want to encrypt the database.
<b>Database storage type</b> (DBStorageType)	General Purpose (SSD)	The database storage type. You can choose <b>General Purpose SSD</b> or <b>Provisioned IOPS</b> . For more information, see the <a href="#">AWS documentation</a> .

### Networking:

Parameter label (name)	Default	Description
<b>Trusted IP range</b> (AccessCIDR)	<i>Requires input</i>	The CIDR IP range that is permitted to access Atlassian products. 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.
<b>Availability Zones</b> (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. If more are Availability Zones are specified, only the first two will be used.
<b>Make instance internet facing</b> (InternetFacingLoad Balancer)	True	Controls whether the load balancer should be visible to the internet (True) or only within the VPC (False).
<b>SSH Key Pair Name *</b> (KeyPairName)	—	The EC2 key pair used by SSH to access the instances. If you don't provide one, the Atlassian Standard Infrastructure stack's key pair is used.

Parameter label (name)	Default	Description
<b>AZ1 private IP address block</b> (PrivateSubnet1CIDR)	10.0.0.0/19	CIDR block for the private subnet located in Availability Zone 1.
<b>AZ2 private IP address block</b> (PrivateSubnet2CIDR)	10.0.32.0/19	CIDR block for the private subnet located in Availability Zone 2.
<b>AZ1 public IP address block</b> (PublicSubnet1CIDR)	10.0.128.0/20	CIDR block for the public (DMZ) subnet located in Availability Zone 1.
<b>AZ2 public IP address block</b> (PublicSubnet2CIDR)	10.0.144.0/20	CIDR block for the public (DMZ) subnet located in Availability Zone 2.
<b>SSL certificate ARN</b> (SSLCertificateARN)	—	The Amazon Resource Name (ARN) of your SSL certificate. The AWS Certificate Manager (ACM) will display the ARN upon certificate creation. If you want to use your own certificate that you generated outside AWS, you must first import it to ACM by following the instructions in the <a href="#">AWS documentation</a> . The import command outputs the ARN.
<b>IP address block for the VPC</b> (VPCCIDR)	10.0.0.0/16	CIDR block for the VPC.

*DNS (optional):*

Parameter label (name)	Default	Description
<b>Existing DNS Name</b> (CustomDnsName)	<i>Optional</i>	An existing custom DNS name for your Jira Data Center instance. This will take precedence over the <b>Route 53 hosted Zone</b> parameter. You must own this domain and configure it to point to the load balancer.
<b>Route 53 Hosted Zone</b> (HostedZone)	<i>Optional</i>	The domain name of the Amazon Route 53 private hosted zone in which to create CNAMEs.

*Application tuning (optional):*

**Note:** For more information about tuning database connections, see the [Jira documentation](#). For more information about Tomcat settings, see the [Apache Tomcat documentation](#).

Parameter label (name)	Default	Description
<b>Tomcat Context Path</b> (TomcatContextPath)	—	The context path of this web application, which is matched against the beginning of each request URI to select the appropriate web application for processing. If used, this path must include a leading forward slash (/).
<b>Catalina options</b> (CatalinaOpts)	—	Java options that are passed to the JVM that runs Jira.
<b>JVM Heap Size Override</b> (JvmHeapOverride)	—	The heap size to use, in MB (e.g., 1024m) or GB (e.g., 1g), to override the default amount of memory to allocate to the JVM for your instance type.
<b>DB Pool Maximum Size</b> (DBPoolMaxSize)	20	The maximum number of database connections that can be opened at any time.
<b>DB Pool Minimum Size</b> (DBPoolMinSize)	20	The minimum number of idle database connections that are kept open at any time.
<b>DB Maximum Idle</b> (DBMaxIdle)	20	The maximum number of database connections that are allowed to remain idle in the pool.
<b>DB Maximum Wait</b> (DBMaxWaitMillis)	10000	The length of time (in milliseconds) that Jira is allowed to wait for a database connection to become available (when there are no free connections available in the pool), before returning an error.
<b>DB Minimum Evictable Idle Time</b> (DBMinEvictableIdleTimeMillis)	180000	The minimum amount of time an object may sit idle in the database connection pool before it is eligible for eviction by the idle object eviction.
<b>DB Minimum Idle Connections</b> (DBMinIdle)	10	The minimum number of idle database connections that are kept open at any time.
<b>DB Remove Abandoned?</b> (DBRemoveAbandoned)	True	Set this parameter to <b>False</b> if you don't want to remove abandoned database connections if they exceed the <b>DB Remove Abandoned Timeout</b> value.
<b>DB Remove Abandoned Timeout</b> (DBRemoveAbandonedTimeout)	60	The length of time (in seconds) that a database connection can be idle before it is considered abandoned.
<b>DB Test On Borrow?</b> (DBTestOnBorrow)	False	Set this parameter to <b>True</b> to test if the database connection is valid when it is borrowed from the database connection pool by Jira.
<b>DB Test While Idle?</b> (DBTestWhileIdle)	True	Set this parameter to <b>False</b> if you don't want to periodically test if the database connection is valid when it is idle.

Parameter label (name)	Default	Description
<b>DB Time Between Eviction Runs</b> (DBTimeBetweenEvictionRunsMillis)	60000	The number of milliseconds to sleep between runs of the idle object eviction thread. When non-positive, no idle object eviction thread will be run.
<b>Enable App to Process Email</b> (MailEnabled)	True	Set this parameter to <b>False</b> if you want to disable mail processing and sending.
<b>Tomcat Accept Count</b> (TomcatAcceptCount)	10	The maximum queue length for incoming connection requests when all possible request processing threads are in use.
<b>Tomcat Connection Timeout</b> (TomcatConnectionTimeout)	20000	The number of milliseconds the Tomcat connector will wait, after accepting a connection, for the request URI line to be presented.
<b>Tomcat Default Connector Port</b> (TomcatDefaultConnectorPort)	8080	The port on which to serve the application.
<b>Tomcat Enable DNS Lookups</b> (TomcatEnableLookups)	False	Set this parameter to <b>True</b> if you want calls to <code>request.getRemoteHost()</code> to perform DNS lookups in order to return the actual host name of the remote client.
<b>Tomcat Maximum Threads</b> (TomcatMaxThreads)	200	The maximum number of request processing threads to be created by the Tomcat connector, which determines the maximum number of simultaneous requests that can be handled.
<b>Tomcat Minimum Spare Threads</b> (TomcatMinSpareThreads)	10	The minimum number of threads always kept running.
<b>Tomcat Protocol</b> (TomcatProtocol)	HTTP/1.1	The protocol for handling incoming traffic.
<b>Tomcat Redirect Port</b> (TomcatRedirectPort)	8443	The port number for Catalina to use when automatically redirecting a non-SSL connector actioning a redirect to an SSL URI.

*AWS Quick Start configuration:*

Parameter label (name)	Default	Description
<b>Quick Start S3 Bucket Name</b> (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.



Parameter label (name)	Default	Description
<b>Quick Start S3 Key Prefix</b> (QSS3KeyPrefix)	quickstart-atlassian-jira/	The <a href="#">S3 key name prefix</a> 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.
<b>ASI identifier</b> (ExportPrefix)	ATL-	Identifier used in all variables (VPCID, SubnetIDs, KeyName) exported from this deployment's Atlassian Standard Infrastructure. Use different identifiers if you're deploying multiple Atlassian Standard Infrastructures in the same AWS Region.

- Parameters for deploying Jira into an existing ASI

[View template](#)

*Jira setup:*

Parameter label (name)	Default	Description
<b>Jira Product *</b> (JiraProduct)	Software	The Jira product to install. You can choose <b>Core</b> (for Jira Core), <b>Software</b> (for Jira Software), or <b>ServiceDesk</b> (for Jira Service Desk).
<b>Version *</b> (JiraVersion)	7.13.3	The version of Jira to install. For a list of versions, see the release notes for <a href="#">Jira Software</a> and <a href="#">Jira Service Desk</a> on the Atlassian website.

*Cluster nodes:*

Parameter label (name)	Default	Description
<b>Amazon CloudWatch Integration</b> (CloudWatchIntegration)	Metrics and Logs	Controls the extent to which Amazon CloudWatch should monitor your deployment. <b>Metrics and Logs</b> monitors node CPU, disk, and network activity, while also performing centralized logging. If the costs incurred by Amazon CloudWatch are an issue, use <b>Metrics Only</b> (to disable logging) or <b>Off</b> (to disable it completely).
<b>Cluster node instance type</b> (ClusterNodeInstanceType)	t3.medium c5.xlarge	The EC2 instance type for the Jira application cluster nodes.
<b>Maximum number of cluster nodes</b> (ClusterNodeMax)	1	The maximum number of Jira nodes in the cluster. When you first launch the Quick Start, leave the default value of 1 node unchanged. In <a href="#">step 3</a> , after configuring Jira for multinode

Parameter label (name)	Default	Description
		clustering, you can change this parameter setting to the desired number of cluster nodes.
<b>Minimum number of cluster nodes</b> (ClusterNodeMin)	1	The minimum number of Jira nodes in the cluster. Leave this parameter at its default setting for a new deployment. You can update it after deployment.
<b>Cluster node instance volume size</b> (ClusterNodeVolumeSize)	50	The size of the cluster node root volume, in GB. You should base this size on the number of application indexes multiplied by 4.
<b>Deployment Automation Git Repository URL</b> (DeploymentAutomationRepository)	<a href="https://bitbucket.org/atlassian/dc-deployments-automation.git">https://bitbucket.org/atlassian/dc-deployments-automation.git</a>	The deployment automation repository to use for per-node initialization. Leave this as default unless you have customizations.
<b>Deployment Automation Branch</b> (DeploymentAutomationBranch)	master	The deployment automation repository branch to pull from.
<b>The Ansible playbook to invoke to initialize the instance</b> (DeploymentAutomationPlaybook)	aws_jira_dc_node.yml	The Ansible playbook to invoke to initialise the Jira node on first start.
<b>Custom command-line parameters for Ansible</b> (DeploymentAutomationCustomParams)	<i>Optional</i>	Additional command-line options for the `ansible-playbook` command. See <a href="https://bitbucket.org/atlassian/dc-deployments-automation/src/master/README.md">https://bitbucket.org/atlassian/dc-deployments-automation/src/master/README.md</a> for more information about overriding parameters.
<b>SSH keyname to use with the repository</b> (DeploymentAutomationKeyName)	<i>Optional</i>	Named Key Pair name to use with this repository. The key should be imported into the SSM parameter store.

### Database:

Parameter label (name)	Default	Description
<b>The database engine to deploy with</b> (DBEngine)	PostgreSQL	Database Engine to use for the application. Choose either PostgreSQL (for Amazon RDS) or Amazon Aurora PostgreSQL.
<b>Database instance class</b> (DBInstanceClass)	db.m5.large	The EC2 instance type for the Amazon RDS database.

Parameter label (name)	Default	Description
<b>RDS Provisioned IOPS</b> (DBIops)	1000	The IOPS for database storage. This value is used only when the <b>Database storage type</b> parameter is set to Provisioned IOPS. Allowed range is 1,000-30,000. The ratio of IOPS to allocated storage must be between 3 and 10.
<b>Master (admin) password *</b> (DBMasterUserPassword)	<i>Requires input</i>	The password for the master (postgres) account. This password must be at least 8 characters and include 1 uppercase, 1 lowercase, 1 number, 1 (non / @ " ') symbol.
<b>Enable RDS Multi-AZ deployment</b> (DBMultiAZ)	True	If <b>True</b> , your database deploys with high availability. When <b>DBEngine</b> equals <b>PostgreSQL</b> , a multi-AZ RDS instance is provisioned. When <b>DBEngine</b> equals <b>Amazon Aurora PostgreSQL</b> , a multi-node Amazon Aurora cluster is provisioned. If high availability isn't a concern, set this parameter to <b>False</b> .
<b>Application user database password</b> (DBPassword)	<i>Requires input</i>	Database password used by Jira. Must be at least 8 characters and include 1 uppercase, 1 lowercase, 1 number, and 1 of the following symbols: ! # \$ % ' * : [ = , ] - _ @ + % &.
<b>Database storage</b> (DBStorage)	200	The storage size, in GB, to allocate to the database.
<b>Database encryption</b> (DBStorageEncrypted)	False	Set this parameter to <b>True</b> if you want to encrypt the database.
<b>Database storage type</b> (DBStorageType)	General Purpose (SSD)	The database storage type. You can choose General Purpose SSD or Provisioned IOPS. For more information, see the <a href="#">AWS documentation</a> .

### Networking:

Parameter label (name)	Default	Description
<b>Make instance internet facing</b> (InternetFacingLoad Balancer)	True	Controls whether the load balancer should be visible to the internet (True) or only within the VPC (False).
<b>Permitted IP range</b> (CidrBlock)	<i>Requires input</i>	The CIDR IP range that is permitted to access Atlassian services. 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.
<b>SSH Key Pair Name *</b> (KeyPairName)	<i>Requires input</i>	The EC2 key pair used by SSH to access the instances.
<b>SSL Certificate ARN</b> (SSLCertificateARN)	<i>Optional</i>	The Amazon Resource Name (ARN) of your SSL certificate. The AWS Certificate Manager (ACM) will display the ARN upon certificate creation. If you want to use your own certificate that you generated outside AWS, you must first

Parameter label (name)	Default	Description
		import it to ACM by following the instructions in the <a href="#">AWS documentation</a> . The import command outputs the ARN.

*DNS (optional):*

Parameter label (name)	Default	Description
<b>Existing DNS Name</b> (CustomDnsName)	—	An existing custom DNS name for your Jira Data Center instance. This will take precedence over the <b>Route 53 hosted Zone</b> parameter. You must own this domain and configure it to point to the load balancer.
<b>Route 53 Hosted Zone</b> (HostedZone)	—	The domain name of the Amazon Route 53 private hosted zone in which to create CNAMEs.

*Application tuning (optional):*

**Note:** For more information about tuning database connections, see the [Jira documentation](#). For more information about Tomcat settings, see the [Apache Tomcat documentation](#).

Parameter label (name)	Default	Description
<b>Tomcat Context Path</b> (TomcatContextPath)	—	The context path of this web application, which is matched against the beginning of each request URI to select the appropriate web application for processing. If used, this path must include a leading forward slash (/).
<b>Catalina options</b> (CatalinaOpts)	—	Java options that are passed to the JVM that runs Jira.
<b>JVM Heap Size Override</b> (JvmHeapOverride)	—	The heap size to use, in MB (e.g., 1024m) or GB (e.g., 1g), to override the default amount of memory to allocate to the JVM for your instance type.
<b>DB Pool Maximum Size</b> (DBPoolMaxSize)	20	The maximum number of database connections that can be opened at any time.
<b>DB Pool Minimum Size</b> (DBPoolMinSize)	20	The minimum number of idle database connections that are kept open at any time.
<b>DB Maximum Idle</b> (DBMaxIdle)	20	The maximum number of database connections that are allowed to remain idle in the pool.
<b>DB Maximum Wait</b> (DBMaxWaitMillis)	10000	The length of time (in milliseconds) that Jira is allowed to wait for a database connection to become available (when there are

Parameter label (name)	Default	Description
		no free connections available in the pool), before returning an error.
<b>DB Minimum Evictable Idle Time</b> (DBMinEvictableIdleTimeMillis)	180000	The minimum amount of time an object may sit idle in the database connection pool before it is eligible for eviction by the idle object eviction.
<b>DB Minimum Idle Connections</b> (DBMinIdle)	10	The minimum number of idle database connections that are kept open at any time.
<b>DB Remove Abandoned?</b> (DBRemoveAbandoned)	True	Set this parameter to <b>False</b> if you don't want to remove abandoned database connections if they exceed the <b>DB Remove Abandoned Timeout</b> value.
<b>DB Remove Abandoned Timeout</b> (DBRemoveAbandonedTimeout)	60	The length of time (in seconds) that a database connection can be idle before it is considered abandoned.
<b>DB Test On Borrow?</b> (DBTestOnBorrow)	False	Set this parameter to <b>True</b> to test if the database connection is valid when it is borrowed from the database connection pool by Jira.
<b>DB Test While Idle?</b> (DBTestWhileIdle)	True	Set this parameter to <b>False</b> if you don't want to periodically test if the database connection is valid when it is idle.
<b>DB Time Between Eviction Runs</b> (DBTimeBetweenEvictionRunsMillis)	60000	The number of milliseconds to sleep between runs of the idle object eviction thread. When non-positive, no idle object eviction thread will be run.
<b>Enable App to Process Email</b> (MailEnabled)	True	Set this parameter to <b>False</b> if you want to disable mail processing and sending.
<b>Tomcat Accept Count</b> (TomcatAcceptCount)	10	The maximum queue length for incoming connection requests when all possible request processing threads are in use.
<b>Tomcat Connection Timeout</b> (TomcatConnectionTimeout)	20000	The number of milliseconds the Tomcat connector will wait, after accepting a connection, for the request URI line to be presented.
<b>Tomcat Default Connector Port</b> (TomcatDefaultConnectorPort)	8080	The port on which to serve the application.
<b>Tomcat Enable DNS Lookups</b> (TomcatEnableLookups)	False	Set this parameter to <b>True</b> if you want calls to <code>request.getRemoteHost()</code> to perform DNS lookups in order to return the actual host name of the remote client.

Parameter label (name)	Default	Description
<b>Tomcat Maximum Threads</b> (TomcatMaxThreads)	200	The maximum number of request processing threads to be created by the Tomcat connector, which determines the maximum number of simultaneous requests that can be handled.
<b>Tomcat Minimum Spare Threads</b> (TomcatMinSpareThreads)	10	The minimum number of threads always kept running.
<b>Tomcat Protocol</b> (TomcatProtocol)	HTTP/1.1	The protocol for handling incoming traffic.
<b>Tomcat Redirect Port</b> (TomcatRedirectPort)	8443	The port number for Catalina to use when automatically redirecting a non-SSL connector actioning a redirect to an SSL URI.

### *AWS Quick Start configuration:*

Parameter label (name)	Default	Description
<b>Quick Start S3 Bucket Name</b> (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.
<b>Quick Start S3 Key Prefix</b> (QSS3KeyPrefix)	quickstart-atlassian-jira/	The <a href="#">S3 key name prefix</a> 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.
<b>ASI identifier</b> (ExportPrefix)	ATL-	Each Atlassian Standard Infrastructure (ASI) uses a unique identifier. If you have multiple ASIs within the same AWS Region, use this field to specify where to deploy Jira.

2. 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**.
3. 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.
4. Choose **Create** to deploy the stack.
5. Monitor the status of the stack. When the status is **CREATE\_COMPLETE**, the deployment is complete.

6. You can use the URL displayed in the **Outputs** tab for the stack to view the resources that were created.

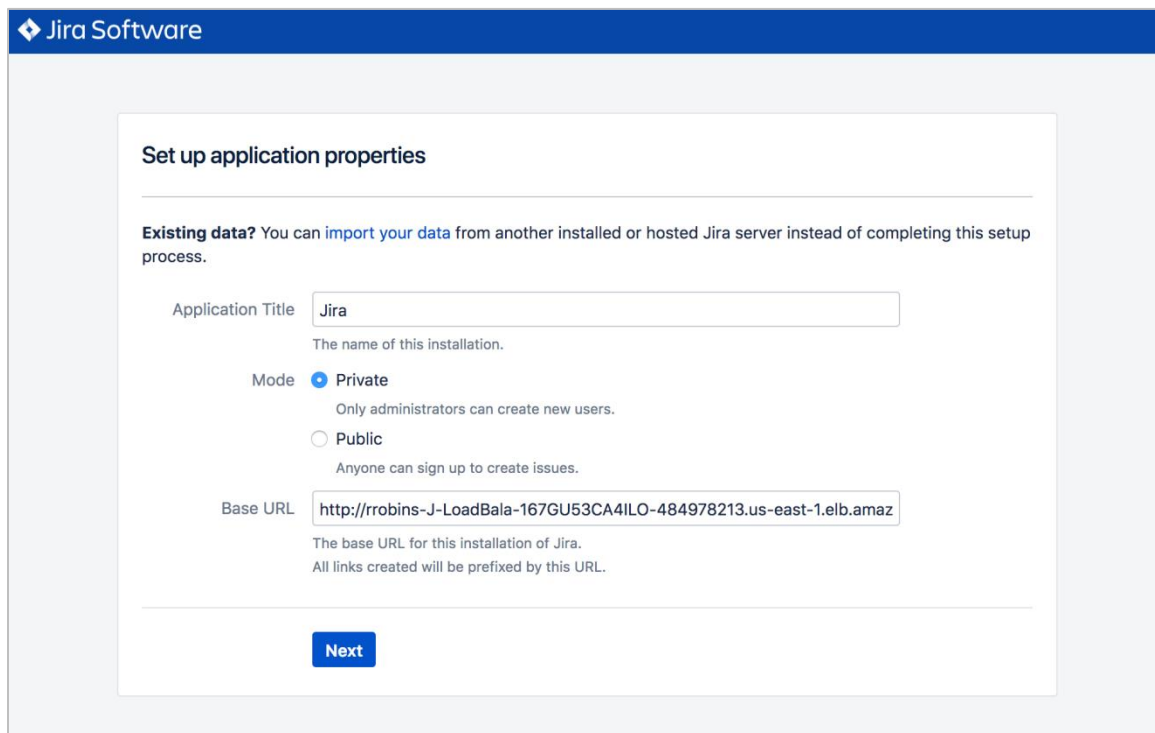
## Step 4. Configure Jira

When you launch the Quick Start, it deploys a single Jira application node (Auto Scaling group of min=1 and max=1 by default).

1. Choose the URL displayed in the **Outputs** tab to go to the Jira setup screen.

**Important:** If you get an [HTTP Error 503 response](#) when you access the URL, it means that the Jira application is still loading. This is expected, and you should wait 2-3 minutes before trying again.

2. On the **Setup application properties** page, enter a title for your Jira application deployment, choose the **Mode** you want, leave the base URL unchanged, and then choose **Next**.



**Set up application properties**

**Existing data?** You can [import your data](#) from another installed or hosted Jira server instead of completing this setup process.

Application Title   
The name of this installation.

Mode ☒ Private  
Only administrators can create new users.

☐ Public  
Anyone can sign up to create issues.

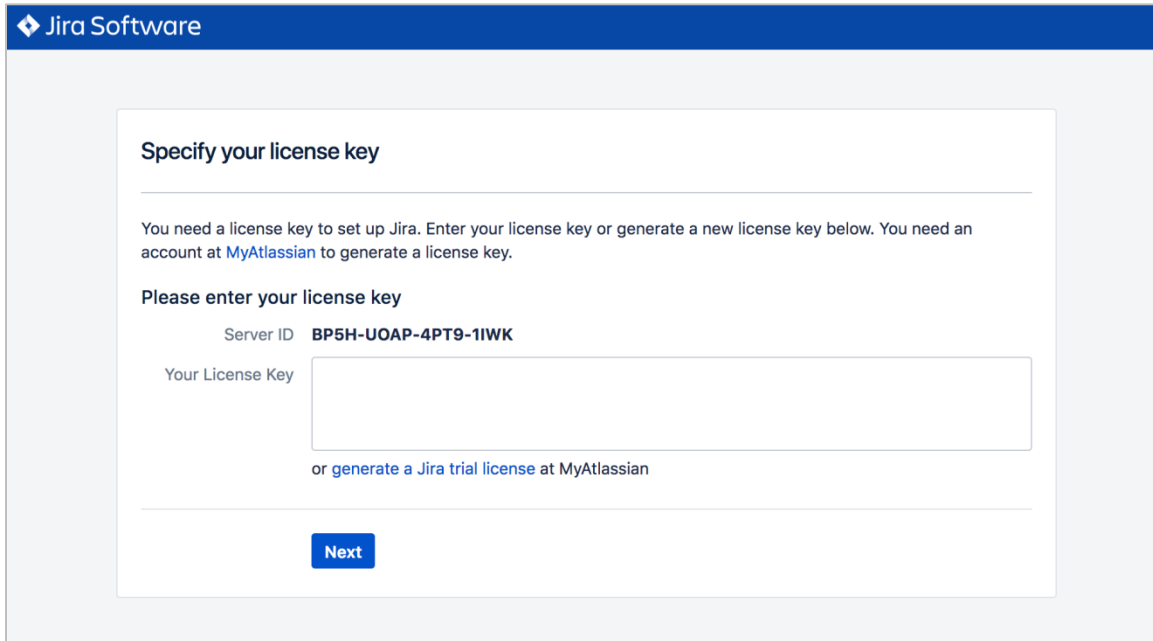
Base URL   
The base URL for this installation of Jira.  
All links created will be prefixed by this URL.

**Next**

**Figure 3: Setting up properties**

3. On the **Specify your license key** page, enter a valid Jira Software or Service Desk Data Center license key. If you don't have a valid license for the Jira application you've selected to deploy, choose **generate a Jira trial license** and sign up for an evaluation

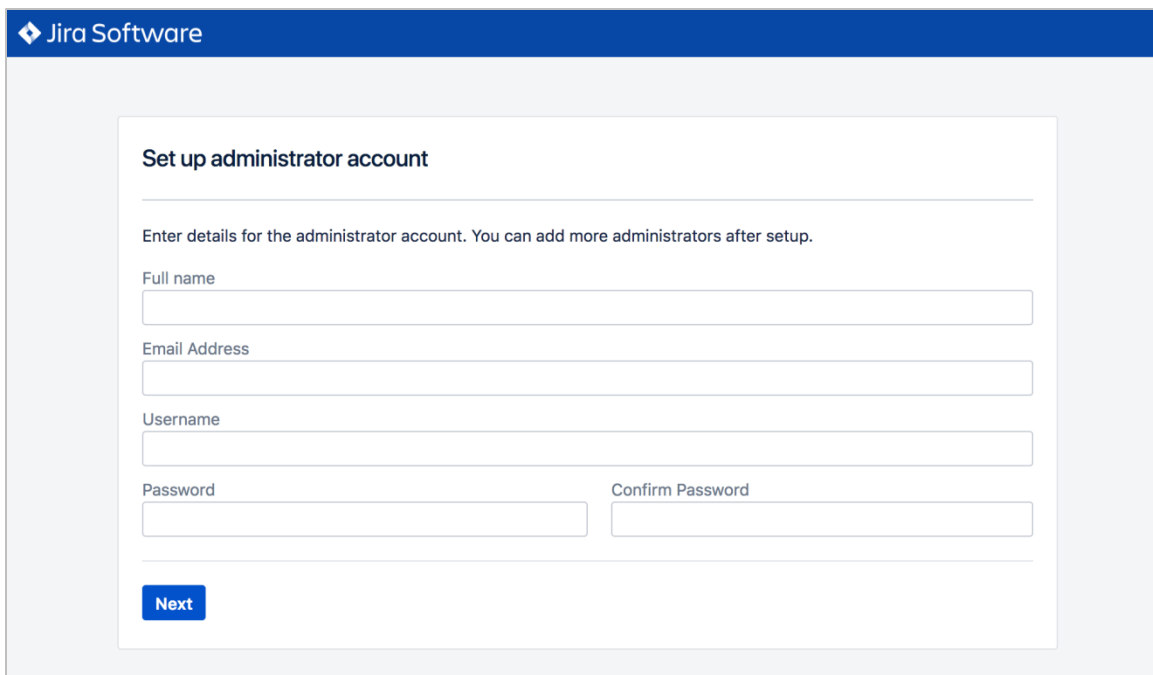
## Data Center license.



The screenshot shows the 'Specify your license key' screen in the Jira Software setup wizard. It has a blue header with the Jira Software logo. The main content area is white and contains the following text: 'Specify your license key', 'You need a license key to set up Jira. Enter your license key or generate a new license key below. You need an account at [MyAtlassian](#) to generate a license key.', 'Please enter your license key', 'Server ID **BP5H-UOAP-4PT9-1IWK**', 'Your License Key' followed by a text input field, and a link 'or generate a Jira trial license at MyAtlassian'. At the bottom is a blue 'Next' button.

Figure 4: Jira licensing

4. To set up the Jira application, you need to create an Administrator account and password. The Administrator account has full access to all data in Jira, so we highly recommend that you choose a strong password for this account. Enter the Administrator's user details in the setup screen (Figure 5), and then choose **Next**.

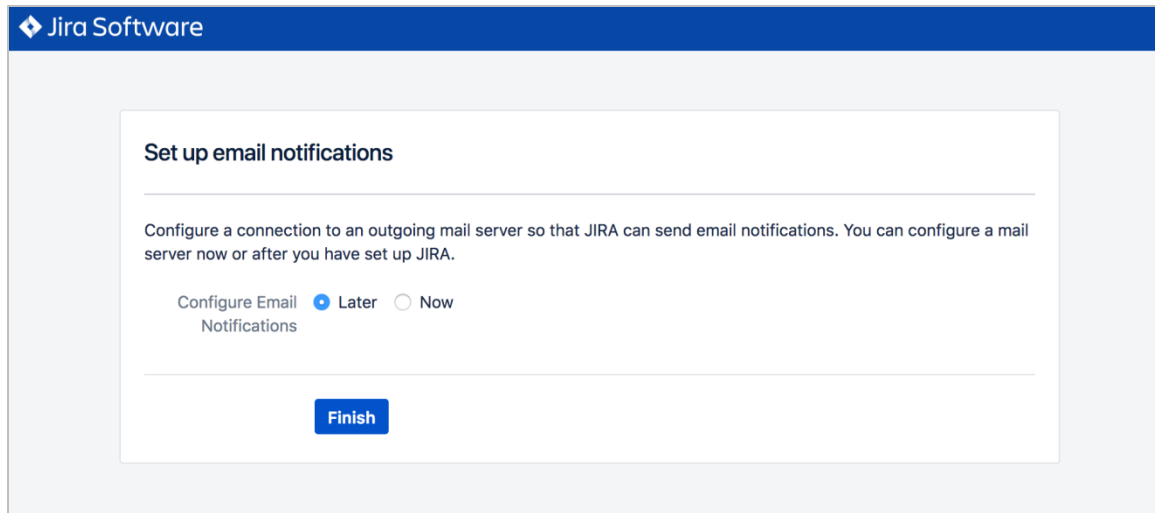


The screenshot shows the 'Set up administrator account' screen in the Jira Software setup wizard. It has a blue header with the Jira Software logo. The main content area is white and contains the following text: 'Set up administrator account', 'Enter details for the administrator account. You can add more administrators after setup.', and five input fields: 'Full name', 'Email Address', 'Username', 'Password', and 'Confirm Password'. At the bottom is a blue 'Next' button.

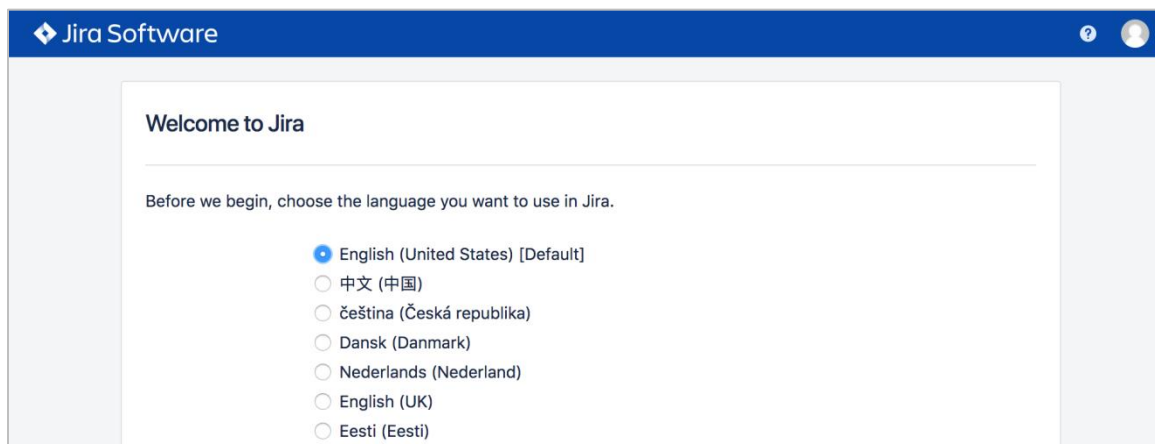


**Figure 5: Jira setup administrator account**

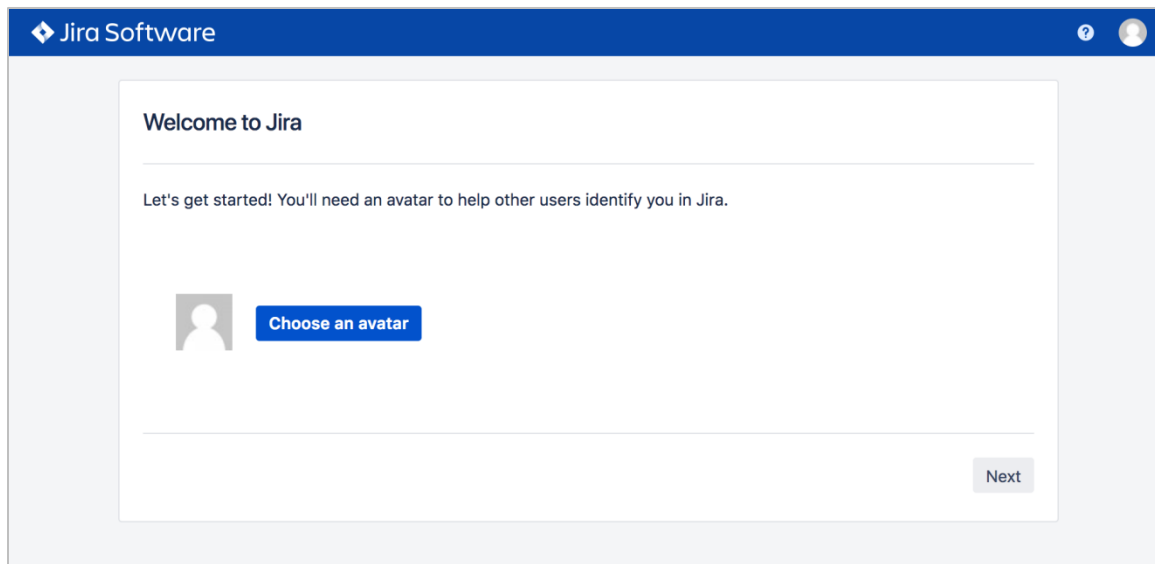
5. On the **Set up email notifications** page, choose **Later**, and then choose **Finish**.

**Figure 6: Email notifications page**

6. In the first **Welcome to Jira** page, choose a language and then choose **Continue**.

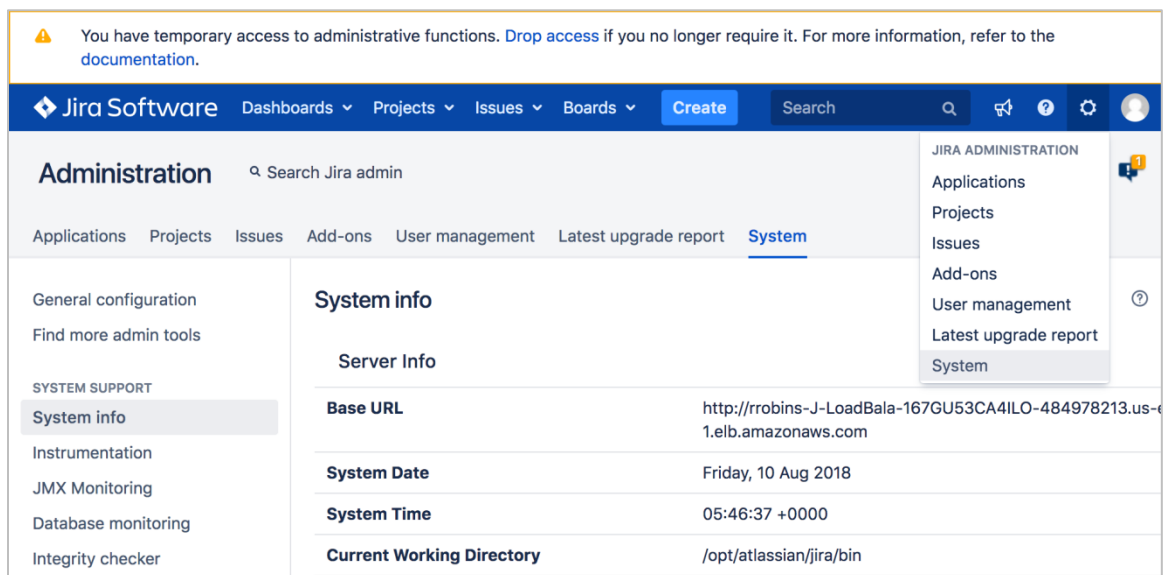
**Figure 7: Choosing a language**

7. In the second **Welcome to Jira** page, choose an avatar for your profile, if you wish, and then choose **Next**.



**Figure 8: Choosing an avatar**

8. On the **Welcome** page, choose **Create sample project**, and enter a name for the project.
9. Choose **Settings** (the gear icon in the upper right), and then choose **System**. You should see a page similar to that illustrated in Figure 9.



**Figure 9: Viewing system info**

10. Scroll down to the **Cluster Nodes** section. You should see your current node in the **Active** state.

File Paths	
Location of Jira Local Home	/var/atlassian/application-data/jira
Location of Jira Shared Home	/media/atl/jira/shared
Location of entityengine.xml	file:/opt/atlassian/jira/atlassian-jira/WEB-INF/classes/entityengine.xml
Location of atlassian-jira.log	/var/atlassian/application-data/jira/log/atlassian-jira.log
Location of indexes	/var/atlassian/application-data/jira/caches/indexes
Location of attachments	/media/atl/jira/shared/data/attachments
Location of backups	/media/atl/jira/shared/export
Cluster nodes	
i-01cef946967db1cf2	<b>Node state:</b> Active <b>Node is alive:</b> true

**Figure 10: Viewing cluster nodes**

Your Jira deployment is now in a state where you can add nodes that will automatically cluster with your existing node.

### Step 5. Add nodes to the cluster

1. Sign in to the AWS Management Console, use the region selector in the navigation bar to choose the AWS Region for your deployment, and open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation/>.
2. Choose the Jira template. From the **Actions** list, choose **Update Stack**.
3. On the **Select Template** page, leave **Use current template** selected, and then choose **Next**.
4. On the **Specify Details** page, in the **Cluster nodes** section of **Parameters**, enter your desired number of cluster nodes in **Minimum number of cluster nodes** and **Maximum number of cluster nodes**. Then, click through to update the stack. This will set a static number of nodes in your cluster.
5. After the stack has finished updating, confirm that the additional nodes have formed a cluster by viewing the **Cluster nodes** section of the **System info** page in Jira.

File Paths	
Location of Jira Local Home	/var/atlassian/application-data/jira
Location of Jira Shared Home	/media/atl/jira/shared
Location of entityengine.xml	file:/opt/atlassian/jira/atlassian-jira/WEB-INF/classes/entityengine.xml
Location of atlassian-jira.log	/var/atlassian/application-data/jira/log/atlassian-jira.log
Location of indexes	/var/atlassian/application-data/jira/caches/indexes
Location of attachments	/media/atl/jira/shared/data/attachments
Location of backups	/media/atl/jira/shared/export
Cluster nodes	
i-01cef946967db1cf2	Node state: Active Node is alive: true
i-0c0e93623aa1d09a7	Node state: Active Node is alive: false

Figure 11: Two nodes in a clustered configuration

Your Jira deployment is now ready to use.

**Note:** You can also migrate Jira data from an existing deployment to this one. For more information, see the [migration guides](#) on the Atlassian website.

## 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. (You'll want to look at the log files in `/var/log/atl.log` and `/var/log/cfn-init.log`.)

**Important:** When you set **Rollback on failure** to **No**, you'll continue to incur AWS charges for this stack. Please make sure to delete the stack when you've finished 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 location we've provided or from another S3 bucket. If you deploy the templates from a local copy on your computer, you might encounter template size limitations when you create the stack. For more information about AWS CloudFormation quotas, see the [AWS documentation](#).

## Additional resources

### AWS services

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

### Atlassian documentation

- Jira Software  
<https://confluence.atlassian.com/jirasoftwareserver/>
- Jira Service Desk documentation  
<https://confluence.atlassian.com/servicedeskserver/>

### Quick Start reference deployments

- Atlassian Standard Infrastructure (ASI) on the AWS Cloud  
<https://fwd.aws/xYyYy>
- Bitbucket Data Center on the AWS Cloud: Quick Start Reference Deployment  
<https://fwd.aws/dEX6W>
- Confluence Data Center on the AWS Cloud: Quick Start Reference Deployment  
<https://fwd.aws/kBpWN>
- AWS Quick Start home page  
<https://aws.amazon.com/quickstart/>

## Git 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.

## Document revisions

Date	Change	In sections
<b>April 2020</b>	Updated the descriptions for parameters <code>DBMultiAZ</code> and <code>DBPassword</code>	<a href="#">Step 3. Review the parameters and create your stack</a>
<b>February 2020</b>	Added Amazon CloudWatch information, added <code>ExportPrefix</code> parameter	<a href="#">Architecture</a> ; <a href="#">Step 3. Review the parameters and create your stack</a>
<b>January 2020</b>	Architecture diagram update	<a href="#">Architecture</a>
<b>October 2019</b>	Added basic Amazon CloudWatch monitoring and logging; added advanced deployment customizations via Ansible	<a href="#">Architecture</a> ; <a href="#">Step 3. Review the parameters and create your stack</a>
<b>August 2019</b>	Added <a href="#">Auto Scaling Groups</a> section; updated parameters	<a href="#">Architecture</a> ; <a href="#">Step 2. Launch the Quick Start</a>
<b>February 2019</b>	Re-architected to deploy into an ASI	Changes in templates and throughout guide
<b>June 2017</b>	Added new <code>CustomDnsName</code> parameter	<a href="#">Step 3</a> parameter tables
<b>November 2016</b>	Added support for Jira Service Desk	Template updates and changes throughout guide
<b>October 2016</b>	Initial publication	—

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