Jv Jive Interactive Intranet

9.x Administrator Guide

Installing Jive

Aurea
Unlimited Possibility

Notices

For details, see the following topics:

Notices

Copyright © 2000–2021. Aurea Software, Inc. (“Aurea”). All Rights Reserved. These materials and all Aurea products are copyrighted and all rights are reserved by Aurea.

This document is proprietary and confidential to Aurea and is available only under a valid non-disclosure agreement. No part of this document may be disclosed in any manner to a third party without the prior written consent of Aurea. The information in these materials is for informational purposes only and Aurea assumes no responsibility for any errors that may appear therein. Aurea reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of Aurea to notify any person of such revisions or changes.

You are hereby placed on notice that the software, its related technology and services may be covered by one or more United States (“US”) and non-US patents. A listing that associates patented and patent-pending products included in the software, software updates, their related technology and services with one or more patent numbers is available for you and the general public’s access at https://markings.ip-dynamics.ai/esw/ (the “Patent Notice”) without charge. The association of products-to-patent numbers at the Patent Notice may not be an exclusive listing of associations, and other unlisted patents or pending patents may also be associated with the products. Likewise, the patents or pending patents may also be associated with unlisted products. You agree to regularly review the products-to-patent number(s) association at the Patent Notice to check for updates.

Aurea and Aurea Software are registered trademarks of Aurea Software, Inc. in the United States and/or other countries. Additional Aurea trademarks, including registered trademarks, are available at: https://www.aurea.com/legal/trademarks/. Jive is a registered trademark of Jive Software, Inc. in the United States and/or other countries. Additional Jive trademarks, including registered trademarks, are available at: https://www.jivesoftware.com/legal/.

Third-party acknowledgments

The following third-party trademarks may appear in one or more Jive guides:

- Amazon is a registered trademark of Amazon Technologies, Inc.
- Apache and Derby is a trademark of Apache Software Foundation.
- Chrome is a trademark of Google Inc.
- Eclipse is a registered trademark of the Eclipse Foundation, Inc.
- HP-UX is a registered trademark of Hewlett-Packard Development Company, L.P.
- IBM, AIX, DB2, and WebSphere are registered trademarks of International Business Machines Corporation.
- Intel and Pentium are registered trademarks of Intel Corporation in the U.S. and/or other countries.
- JBoss is a registered trademark, and CentOS is a trademark, of Red Hat, Inc. in the U.S. and other countries.
- Linux is a registered trademark of Linus Torvalds.
- Microsoft, Active Directory, Internet Explorer, SharePoint, SQL Server, Visual Studio, and Windows are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Mozilla and Firefox are registered trademarks of the Mozilla Foundation.
- Oracle and Java are registered trademarks of Oracle and/or its affiliates.
- Progress and OpenEdge are registered trademarks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and other countries.
- Red Hat and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the U.S. and other countries.
- SAP and SAP NetWeaver are registered trademarks of SAP SE in Germany and in several other countries.
- SUSE is a registered trademark of SUSE, LLC.
- Ubuntu is a registered trademark of Canonical Limited in the United States and/or other countries.
- UNIX is a registered trademark of The Open Group in the United States and other countries.

All other marks contained herein are for informational purposes only and may be trademarks of their respective owners.

Table of Contents

Aurea global support

If you encounter a problem while using an Aurea product or require assistance with downloading the software or upgrading a product release, please, try to:

- Search the articles on the Aurea Knowledge Base for solutions to your issues.
- Search the product documentation and other product-related information that are also available on Support Central.

If you still cannot find a solution, open a ticket on Aurea Support Central. Information about the support organization is available on Support Portal as well.

You can also find the setup files on Support Portal.

For information about purchasing an upgrade or professional services, contact your account executive. If you do not know who your account executive is, or for other queries, contact us through our website.

1

Installing

You can use the following instructions to prepare for and complete an on-premise Jive installation. Jive production installations require multiple servers, a database, and expert knowledge of your enterprise systems; for details, refer to System requirements. Note that these instructions are a template and your requirements will be specific to your community. You should also have a plan to back up your installation, as described in Backup and storage considerations on page 50.

Note: If you want to do a quick install for evaluation or development purposes, see Quick installation for Jive evaluation on page 58.

If you are installing any of Jive's add-ons, such as Jive for Outlook or Jive Daily Hosted, check the documentation for their separate system and installation requirements at https://docs.jivesoftware.com/.

This section includes information on how to install the pieces that make up the Jive platform.

For details, see the following topics:

- CLI installer overview
- Startup property reference
- Connecting through proxy server
- Preparing to connect to Jive Hosted services
- Managing client certificates
- Installation overview
- CLI installation tasks
- Setting up Document Conversion
- Setting up cache server
- Setting up conversion machine
- Post-installation tasks
- Installing on cluster
- Adding features with plugins
- Configuring delegated authentication
- Configuring binary storage provider
- Backup and storage considerations
- Configuring application with the Setup wizard
- Quick installation for Jive evaluation

CLI installer overview

The Jive command-line interface (CLI) installer provides a better way to start, stop and configure an installation, and gives you more flexibility.

The command line procedure provides a more centralized interface for running, monitoring, and troubleshooting your Jive installation. The installation also provides interactive feedback that helps you understand whether you're meeting installation and upgrade requirements and what your next steps should be. It also records your startup modifications so you can be aware of your customizations during future upgrades.

Features of the Jive CLI installer and interface

Saves time tracking down configurations	The site-specific configurations are in one place so you can easily view and override them.

Provides tools for managing the platform We provide a single command-line interface for starting and stopping services and configuring the startup environment.

Enables separation of roles in the IT environment If your organization prefers to separate the root administration and application administration roles, you can install the application in a custom location other than the default `/usr/local/jive`, and run commands using a different user account than the default `jive` user. The root user is not necessary for the routine operation of the platform.

Startup property reference

The following tables list the startup properties available for your Jive installation, grouped by the service they support.

For information on how to set these properties, see Startup property commands.

Table 1: Main startup properties

Main startup property name	Description	Default
`main.gc_log_file_size`	The maximum size of each java gc log file (MB) before rotation	`1024`
`main.java_bin`	Path to the java binary	`{{main.java_home}}/bin/java`
`main.java_home`	Directory where java is installed	`{{main.jive_home}}/java`
`main.jive_group`	Primary group for the user account that runs the Jive platform	`jive`
`main.jive_home`	Directory where Jive is installed	`/usr/local/jive`
`main.jive_user`	User account for running the Jive platform	`{{main.jive_home}}/var/logs`
`main.log_dir`	Directory for all log files	
`main.number_of_gc_log_files`	The number of java gc log file each service should retain before overwriting	`2`
`main.pidfile_dir`	Directory where all process ID files are stored	`{{main.jive_home}}/var/run`
`main.skip_setup`	Always skip setup warnings when starting services	`False`

Main startup property name	Description	Default
`main.start_wait_seconds`	Number of seconds to wait after startup before verifying that the process is still running	`5`
`main.system_info_cmd`	Command for capturing overall system info when running 'jive snap'./usr/bin/top -n1 -b -H	
`main.tomcat_home`	Directory where tomcat is installed	`{{main.jive_home}}/tomcat`

Table 2: Cache startup properties

Cache startup property name	Description	Default
`cache.admin_max_threads`		`60`
`cache.admin_port`	The port on which cache service is run	`6667`
`cache.custom_jvm_args`	Used for additional custom java arguments for cache	
`cache.enabled`	Always start cache when `jive start` is run without arguments	`False`
`cache.hostnames`	The short-form, unqualified hostname of the machine on which the cache service is running. For configurations with multiple cache nodes, this can be a comma-separated list of cache node hostnames. In this case, the value for `cache.hostnames` must be the same on each cache node (for example, same order)	
`cache.jmxremote_port`	JMX port for the cache service	`6650`
`cache.jvm_heap_max`	Java -Xmx value for cache, in MB.	`2048`
`cache.jvm_heap_min`	Java -Xms value for cache, in MB	`2048`
`cache.max_threads`		`100`

Cache startup property name	Description	Default
`cache.priority`	Startup priority order for cache with respect to other jive services on this machine	`0`
`cache.snap_cmd`	Command for capturing performance information from cache when running `jive snap`	`{{main.java_home}}/bin/jstack -l $PID && {{main.java_home}}/bin/jmap -histo $PID | /usr/bin/head -n 103`
`cache.socket_port`	The port on which cache service is run	`6666`
`cache.stdout`	File to send cache's STDOUT stream	`{{main.log_dir}}/cache-service.out`

Table 3: DocConverter startup properties

DocConverter startup property name	Description	Default
`docconverter.custom_jvm_args`	Any additional custom java arguments for DocConverter go here	
`docconverter.enabled`	Always start DocConverter when `jive start` is run without arguments	`False`
`docconverter.http_addr`	IP for DocConverter to listen on	`0.0.0.0`
`docconverter.http_monitor_addr`	Monitor IP for DocConverter	`127.0.0.1`
`docconverter.http_monitor_port`	Monitor port for DocConverter	`19005`
`docconverter.http_port`	HTTP port for DocConverter	`19003`
`docconverter.imager_path`	Path to the imager utility	`{{main.jive_home}}/bin/imager/bin/imager`
`docconverter.jvm_heap_max`	Java `-Xmx` value for DocConverter, in MB	`256`
`docconverter.jvm_heap_min`	Java `-Xms` value for DocConverter, in MB	`256`
`docconverter.jvm_heap_perm (DEPRECATED)`	Java `-XX:MaxPermSize` value for DocConverter, in MB	`64`

DocConverter startup property name	Description	Default
`docconverter.kill_one_oo_path`	Path to the script which allows DocConverter to stop its running instances of OpenOffice	`{{docconverter.service_home}}/bin/kill_one_oo.sh`
`docconverter.openoffice_path`	Path to the OpenOffice binary	`{{main.jive_home}}/opt/libreoffice4.1/program/soffice.bin`
`docconverter.pdf2swf_path`	Path to the pdf2swf utility	`{{main.jive_home}}/bin/pdf2swf`
`docconverter.priority`	Startup priority order for DocConverter service with respect to other jive services on this machine	`0`
`docconverter.server_port`	Server port for DocConverter service	`19004`
`docconverter.service_home`	Base directory for the DocConverter	`{{main.jive_services_home}}/docconverter`
`docconverter.snap_cmd`	Command for capturing performance information from DocConverter when running `jive snap`	`{{main.java_home}}/bin/jstack -l $PID && {{main.java_home}}/bin/jmap -histo $PID | /usr/bin/head -n 103`
`docconverter.stdout`	File to send DocConverter's STDOUT stream	`{{main.log_dir}}/docconverter.out`
`docconverter.textextract_path`	Path to the textextract utility	`{{docconverter.service_home}}/utils/bin/text_extract`

Table 4: Activity engine startup properties

Activity engine startup property name	Description	Default
`eae.custom_classpath_additions`	Custom paths for eae's CLASSPATH environment variable can be included here (for example, Oracle database driver install paths). Colon-separated list	
`eae.custom_jvm_args`	Any additional custom java arguments for eae go here	

Activity engine startup property name	Description	Default
`eae.custom_ld_library_path_additions`	Custom paths for eae's LD_LIBRARY_PATH environment variable can be included here (for example, Oracle database driver install paths). Colon-separated list	
`eae.enabled`	Always start eae when `jive start` is run without arguments	`False`
`eae.jmx_port`	Port on which JMX is run	`8026`
`eae.jvm_heap_max`	Java `-Xmx` value for eae, in MB	`2048`
`eae.jvm_heap_min`	Java `-Xms` value for eae, in MB	`2048`
`eae.priority`	Startup priority order for eae with respect to other jive services on this machine	`0`
`eae.service_home`	Base directory for the Activity Engine service	`{{main.jive_services_home}}/eae-service`
`eae.snap_cmd`	Command for capturing performance information from eae when running `jive snap`	`{{main.java_home}}/bin/jstack -l $PID && {{main.java_home}}/bin/jmap -histo $PID | /usr/bin/head -n 103`
`eae.stdout`	File to send eae's STDOUT stream	`{{main.log_dir}}/eae-service.out`

Table 5: Httpd startup properties

Httpd startup property name	Description	Default
`httpd.apachectl_cmd`	The base command by which apachectl is invoked for httpd	`{{main.jive_home}}/httpd/bin/apach`
`httpd.apachectl_lock_file`	Path to httpd's apachectl lock file	`{{main.pid-file_dir}}/jive-httpd-accept.lock`
`httpd.apachectl_pid_file`	Path to httpd's apachectl pid file.	`{{main.pid-file_dir}}/jive-httpd.pid`
`httpd.conf`	Path to the main httpd config file	`{{main.jive_home}}/etc/httpd/conf/`

Httpd startup property name	Description	Default
`httpd.enabled`	Always start httpd service when `jive start` is run without arguments	`False`
`httpd.lang`	Locale for httpd (HTTPD_LANG)	`C`
`httpd.port`	Port on which httpd listens for normal HTTP connections	`8080`
`httpd.priority`	Startup priority order for httpd with respect to other jive services on this machine	`2`
`httpd.procname`	The name under which a running httpd service appears in the process list	`{{main.jive_home}}/httpd/bin/jive-httpd`
`httpd.server_admin_email`	Admin email for httpd	`support@jivesoftware.com`
`httpd.ssl_certificate_file`	Path to SSL cert when `httpd.ssl_enabled` is set to `True`	
`httpd.ssl_certificate_key_file`	Path to SSL key when `httpd.ssl_enabled` is set to `True`	
`httpd.ssl_enabled`	Enable SSL for httpd	`False`
`httpd.ssl_port`	Port on which httpd accepts ssl-enabled connections when `httpd.ssl_enabled` is set to `True`	`8443`
`httpd.stdout`	File to send httpd's STDOUT stream	`{{main.log_dir}}/jive-httpd.out`

Table 6: Ingress-replicator startup properties

Ingress-replicator startup property name	Description	Default
`ingress-replicator.custom_jvm_args`	Any additional custom java arguments for ingress-replicator go here	
`ingress-replicator.debug_port`	Search debug port	`29001`
`ingress-replicator.enabled`	Always start ingress-replicator service when `jive start` is run without arguments	`False`

Ingress-replicator startup property name	Description	Default
ingress-replicator.jmx_port	Search JMX port	29002
ingress-replicator.jvm_heap_max	Java -Xmx value for ingress-replicator, in MB	2048
ingress-replicator.jvm_heap_min	Java -Xms value for ingress-replicator, in MB	2048
ingress-replicator.on_out_of_memory_error	Command to run on out-of-memory error.	echo "Out of memory crash!"
ingress-replicator.priority	Startup priority order for ingress-replicator with respect to other jive services on this machine	0
ingress-replicator.service_home	Base directory for the ingress-replicator service	{{main.jive_home}}/services/ingress-replicator-service
ingress-replicator.snap_cmd	Command for capturing performance information fromingress-replicatorwhen running jive snap	{{main.java_home}}/bin/jstack -l $PID && {{main.java_home}}/bin/jmap -histo $PID | /usr/bin/head -n 103
ingress-replicator.stdout	File to send ingress-replicator's STDOUT stream	{{main.log_dir}}/ingress-replicator/stdout.log

Table 7: Search startup properties

Search startup property name	Description	Default
search.custom_jvm_args	Any additional custom java arguments for search go here	
search.debug_port	Search debug port	27001
search.enabled	Always start search service when jive start is run without arguments	False
search.jmx_port	Search JMX port	27002
search.jvm_heap_max	Java -Xmx value for search, in MB	2048
search.jvm_heap_min	Java -Xms value for search, in MB	2048
search.on_out_of_memory_error	Command to run on out-of-memory error	echo "Out of memory crash!"

Search startup property name	Description	Default
`search.priority`	Startup priority order for search with respect to other jive services on this machine	`0`
`search.service_home`	Base directory for the search service	`{{main.jive_home}}/services/search-service`
`search.snap_cmd`	Command for capturing performance information from search when running `jive snap`	`{{main.java_home}}/bin/jstack -l $PID && {{main.java_home}}/bin/jmap -histo $PID | /usr/bin/head -n 103`
`search.stdout`	File to send search's STDOUT stream	`{{main.log_dir}}/search/stdout.log`

Table 8: Webapp startup properties

Webapp startup property name	Description	Default
`webapp.app_cluster_jvmroute`	A string that is added to the JSESSIONID cookie served by the webapp, to allow cookie-based load balancing	
`webapp.appname`	Webapp instance name	
`webapp.context`	URL context for the webapp. For example, if your site is `http://my.company.com/community`, `community` is the context	blank
`webapp.custom_classpath_additions`	Custom paths for webapp's CLASSPATH environment variable can be included here (for example, Oracle database driver install paths). Colon-separated list	
`webapp.custom_jvm_args`	Any additional custom java arguments for the webapp go here	
`webapp.custom_ld_library_path_additions`	Custom paths for webapp's LD_LIBRARY_PATH environment variable can be included here (for example, Oracle database driver install paths). Colon-separated list	

Webapp startup property name	Description	Default
webapp.enabled	Always start webapp service when jive start is run without arguments	False
webapp.http_addr	IP for the webapp's main Tomcat connector. This is set to localhost because the jive httpd service must always serve as a proxy	127.0.0.1
webapp.http_maxthreads	The maxThreads attribute for the webapp's main Tomcat connector	15
webapp.http_monitor_addr	IP for the webapp's monitoring Tomcat connector	127.0.0.1
webapp.http_monitor_maxthreads	The maxThreads attribute for the webapp's monitoring Tomcat connector	15
webapp.http_monitor_port	Port for the webapp's monitoring Tomcat connector	9002
webapp.http_monitor_proxy_name	The webapp's monitoring Tomcat connector's proxyName attribute	
webapp.http_monitor_proxy_port	The webapp's monitoring Tomcat connector's proxyPort attribute	{{webapp.http_monitor_port}}
webapp.http_monitor_proxy_scheme	The webapp's monitoring Tomcat connector's scheme attribute	http
webapp.http_port	Port for the webapp's main Tomcat connector	9001
webapp.http_proxy_name	The webapp's main Tomcat connector's proxyName attribute. This is your external load balancer's address	
webapp.http_proxy_port	The webapp's main Tomcat connector's proxyPort attribute. This is your external load balancer's port.	{{httpd.port}}
webapp.http_proxy_scheme	The webapp's main Tomcat connector's scheme attribute, http or https. Choose based on the type of connection provided by your external load balancer	http

Webapp startup property name	Description	Default
webapp.jive_application	Directory containing the webapp application files	{{webapp.jive_base}}/application
webapp.jive_base	Base directory for the web application	{{main.jive_home}}/applications/{{webapp.appname}}
webapp.jive_instance_home	Webapp home directory	{{webapp.jive_base}}/home
webapp.jmx_enabled	Enable JMX on the webapp	False
webapp.jmx_port	Port on which to run JMX, if webapp.jmx_enabled is set to True.	6651
webapp.jvm_heap_max	Java -Xmx value for webapp, in MB	2048
webapp.jvm_heap_min	Java -Xms value for webapp, in MB	2048
webapp.jvm_heap_perm (DEPRECATED)	Java -XX:MaxPermSize value for webapp, in MB	512
webapp.log4j_conf	The path to a minimal log4j.properties file used prior to initializing the application	{{webapp.jive_base}}/conf/log4j.properties
webapp.priority	Startup priority for webapp with respect to other jive services on this machine	1
webapp.server_port	Webapp server port	9000
webapp.snap_cmd	Command for capturing performance information from webapp when running jive snap	{{main.java_home}}/bin/jstack -l $PID && {{main.java_home}}/bin/jmap -histo $PID | /usr/bin/head -n 103
webapp.stdout	File to send webapp's STDOUT stream	{{main.log_dir}}/{{webapp.appname}}.out
webapp.stop_cmd_jvm_heap_max	Java -Xmx value for the webapp stop command, in MB	512
webapp.stop_cmd_jvm_heap_min	Java -Xms value for the webapp stop command, in MB	512

Webapp startup property name	Description	Default
`webapp.stop_cmd_jvm_heap_perm` (DEPRECATED)	Java `-XX:MaxPermSize` value for the webapp stop command, in MB	
`webapp.work_dir`	Temp file directory for webapp	`{{main.jive_home}}/var/work/{{webapp.appname}}`

Connecting through proxy server

Certain core components and features require Jive to access information from across the firewall. If you use a proxy server to access the Internet, setting up the proxy connection using the Proxy Server settings in the Admin Console ensures you can connect to Jive Daily Hosted. You can also set exceptions if you want to connect to certain hosts directly, rather than through the proxy server.

Fastpath: Admin Console: System > **Settings** > **Proxy Server**

Setting JVM properties for Recommender

Connecting to the Recommender service requires setting several JVM properties after you complete the Activity Engine installation. To set the required properties:

1. Use `jive set eae.custom_jvm_args` to add the required java args to the service invocation:

   ```
   jive set eae.custom_jvm_args -Dhttp.proxyHost=webproxy_address
   -Dhttp.proxyPort=webproxy_port -Dhttp.proxyUser=webproxy_user
   -Dhttp.proxyPassword=webproxy_pwd
   ```

   For example, add:

   ```
   -Dhttp.proxyHost=webproxy.eng.jiveland.com -Dhttp.proxyPort=3128
   -Dhttp.proxyUser=jive -Dhttp.proxyPassword=jive
   ```

   The `proxyUser` and `proxyPassword` properties are required. If your proxy server does not require a user name and password, include these options with blank values.

   : If you have already set `eae.custom_jvm_args`, you need to append the existing setting to the string above, or it will be overwritten.

2. If the system is NOT the production instance, go to **System** > **Management** > **System Properties** and set the system property `jive.eae.instance.type` to the value `2`.
3. Restart the Activity Engine instance and the application servers.

Advanced proxy configurations

If you have DNS proxying enabled, you also need to set the `jive.apps.proxy.whitelist.cidrs` system property to include the IP address (single node) or CIDR addresses of your proxy server. Domain names are not supported. You can separate multiple addresses with spaces.

If your site uses client certificate validation, you should contact Support for assistance.

Reverse proxying can modify data in ways that are not compatible with running Jive. Reverse proxies are often configured to reject GETs with special characters in the query string, strip the bodies from PUT commands, and add prefixes to cookie names. For example, Jive Apps require the colon (:) character to be supported in URLs. If you use IIS-based reverse proxying with .NET 4.5 or earlier, colons are blocked by default. In this case, upgrading to .NET 5.0 resolves the problem.

Preparing to connect to Jive Hosted services

To ensure a successful installation, you need to make sure you can connect successfully to several component services that are hosted by Jive.

In addition to the Core components of Jive that you install on your own servers, running Jive requires connections through your firewall to the Recommender service, and, if you use them, the Mobile Gateway, Search and Video services. The following sections provide the ports and addresses you use when ensuring your firewall can access the correct ports and domains for these services.

General best practices

For all these components, you need to ensure that your proxy server is configured to access resources outside the firewall. For more information, see Connecting through proxy server on page 18.

Jive Core components

Component	Jive CLI name	Ports	Direction	Domains or IPs

Jive Core (Hosted only) when connecting to custom services behind a firewall[1]				AWS US-East-1 Region:
				`34.192.45.122, 34.198.91.162, 34.231.78.214, 34.225.172.123, 34.193.143.104, 52.55.123.87, 52.20.222.9, 34.230.231.2, 34.197.60.63, 52.207.30.159, 3.213.1.211`
				AWS EU-West Region:
				`54.154.171.198, 108.129.50.14, 52.31.199.172, 34.247.7.187, 34.252.244.183, 52.211.222.108, 63.33.30.202`

Recommender service

Make sure your Activity Engine servers can connect through the firewall using the following settings.

Component	Jive CLI name	Ports	Direction	Domains or IPs
Activity Engine	eae	TCP port: `7020` JMX port: `7021,8026` RMI ports: `33030,56844`	Open	

After installation is complete, you can check the status of each Activity Engine's connection to the Recommender service by using the Activity Engine page in the Admin Console.

- Go to **System** > **Settings** > **Activity Engine** and check the Recommender column of the Activity Engine Overview table.

Note that if you relocate an instance by changing the `jiveURL`, and you then enable and disable the Recommender on that instance, the Recommender restarts with a new ID for the instance. Recommendations from before the `jiveURL` changed will be lost.

Search service

With an on-premise installation, you can choose from either Jive Cloud Search or On-Premise Search.

The Jive Cloud Search offers socially contextual search and requires a simple support case to whitelist your IP Addresses. Jive Cloud and Hosted both use Cloud Search. For more information, see Cloud Search service.

On-Premise Search uses the On-Premise search service installed in your Jive environment. For more information, see On-Premise Search service.

The following table shows which ports and domains are required so the web application nodes can contact the Search service.

Component	Jive CLI name	Ports	Direction	Domains or IPs

Cloud Search service		`443`	Outbound	US customers only:
				`search-ingress-adapter.aws-us-east-1-prod.svc.jivehosted.com`
				`search-query.aws-us-east-1-prod.svc.jivehosted.com`
				EU customers only:
				`search-ingress-adapter.aws-eu-west-1-prod.svc.jivehosted.com`
				`search-query.aws-eu-west-1-prod.svc.jivehosted.com`

On-Premise Search service	search-service	Service port: `30000` Debug port: `27001` JMX port: `27002`	Open	`localhost`

[1] *This applies to Hosted communities only. If your organization utilizes whitelisting to connect your Jive instance to your organization's services behind a firewall, you need to whitelist the IPs as specified. This may be required, for example, if your Jive community is configured for LDAP directory syncing, and you utilize a whitelist on your firewall to allow your Jive community to connect to it.*

You can check the status and connection to the Search service using the Search page in the Admin Console.

- Go to **System** > **Settings** > **Search** and check the Search service settings.

You can also use the Admin Console's Search page to make changes to the location or type of Search service. If you change the type of search service (Cloud Search or On-Premise Search), you need to perform a full index rebuild. For more information, see Selecting Search service.

Mobile Push service (optional)

If you plan to have your community members receive Jive push notifications, you need access to the Jive Mobile Push service. Make sure the following ports and IPs are enabled so the web application nodes can contact it.

Component	Ports	Direction	Domains or IPs
Mobile (all locations, including EMEA) when you are sending push notifications to the publicly available apps	`443`	Outbound from Jive instance	`mobile-push.prod.jiveon.com` (`204.93.64.255` and `204.93.64.252`)
Mobile (all locations, including EMEA) when you are using your custom branded iOS app with your own push notification certificate	TCP port `5223` (used by devices to communicate to the APNs servers) TCP port `2195` (used to send notifications to the APNs) TCP port `2196` (used by the APNs feedback service) TCP Port `443` (used as a fallback on Wi-Fi only, when devices are unable to communicate to APNs on port `5223`)	Outbound from Jive instance	
Mobile (all locations, including EMEA) when you are using your custom branded Android app with your own Google Cloud Messaging key	`443`	Outbound from Jive instance	`https://android.googleapis.com/gcm/se`

Video service

Setting up the firewall for video is complex. For more information, see Configuring video.

Managing client certificates

If your corporate network uses client certificates for authentication, you need to configure Jive so it can authenticate properly. You can use the Admin Console to choose how to handle client certificates. If you choose a method that allows for storing certificates in Jive, you can also view and manage them in the Admin Console. Jive interface accepts PKCS12 encrypted keys, just as your browser does. Uploading a new certificate and key takes effect immediately, and all the nodes in a cluster share the same keystore.

Fastpath: Admin Console: System > **Settings** > **Client Certificates**

Certificate management strategies

- Select **Java** as the client certificate strategy if you're planning to use Java's keytool to manage certificates outside Jive using Java system properties. The Client Certificates dialog box then becomes read-only, and you can't upload any certificates. For more information about the Java keytool, refer to the Oracle documentation.
- Select **Issuer** as the client certificate strategy if you want to select certificates by issuer. You are required to provide the key and the certificate password when you upload your client certificates to the Jive keystore.
- Select **Domain** as the client certificate strategy if you want to specify the certificates to use for a specific domain. You are required to provide the domain and the certificate password used to access the cert file when you upload your client certificates to the Jive keystore.

Testing certificate validation

If you put a test URL in the box and click **Test Connectivity**, you get a detailed report on client and server connectivity.

Installation overview

When you install the Jive platform, you perform several steps in a specific sequence. Here is an overview of all the installation steps and the order in which you should perform them

Step	What you're installing	Required or optional	Install instructions
1	—	Required	Review the following topics to understand a typical configuration, required components and sizes, and how to scale your system: • Jive Enterprise architecture • System requirements • RPM dependencies by operating system • Pre-installation requirements as root on CLI • Network requirements
2	—	Required	Review the information in Preparing to connect to Jive Hosted services on page 19 to ensure your servers can communicate with the externally hosted components. If you use a proxy server, make sure it's configured correctly to access the required external resources. For more information, see Connecting through proxy server on page 18.
3	Activity Engine Database, Core Databases,Analytics Database	Required	Install the Activity Engine, Core, and Analytics databases each on their own server, as described in Database configuration and best practices, and create users on them.
4	Activity Engine	Required	Install the Jive Linux package on your Activity Engine node, as described in Installing Jive package and starting up on page 27. This node should be able to access the Recommender Service, as described in Preparing to connect to Jive Hosted services on page 19.
5	Search	Required for On-Premise Search only	If you are using On-Premise Search, install the Jive Linux package on your Search node, as described in Installing Jive package and starting up on page 27. Make sure the web app node can access the Search service, as described in Search service on page 21.

Step	What you're installing	Required or optional	Install instructions
6	Jive Web Apps	Required	Install the Jive Linux package on your web app nodes, as described in Installing Jive package and starting up on page 27. Any web app nodes need access to the domain for Mobile Gateway, directly or through your proxy server, as described in Preparing to connect to Jive Hosted services on page 19 and Connecting through proxy server on page 18.
7	Configure Cache Server	Required	Install the Jive Linux package on the cache servers, as described in Installing Jive package and starting up on page 27. For more information on a typical configuration and how to scale your system, see Jive Enterprise architecture.
8	Set up First Web App Using Setup Wizard	Required	With a supported web browser, navigate to `http://<hostname>/`, where `hostname` is the DNS-resolvable name of the server where you installed the Jive application on your primary web application node. There, you are prompted to finish configuring the Jive application via the Admin Console Setup wizard.
9	Select Correct Search Type	Required	In the Other Settings page of the Setup wizard, choose between the Cloud and On-Premise Search service. If you're using On-Premise Search, make sure you select **On-Premise Search**. If you're using Cloud Search, contact Jive support to get your IP addresses whitelisted.
10	Configure Cluster Nodes	Required	Configure your cluster. Configurations require a clustered environment unless you are only evaluating Jive, as described in Quick installation for Jive evaluation on page 58. To configure a clustered environment, you need to install the Jive Linux package on each node of the cluster and configure it appropriately, as described in Installing Jive package and starting up on page 27 and Configuring cluster node (optional) on page 45.

Step	What you're installing	Required or optional	Install instructions
11	Document Conversion Module	Required	Install the Jive Linux package on the document conversion node, as described in Installing Jive package and starting up on page 27. This module allows documents to be previewed in Jive without modifying their original format. For more information, and to complete the required server configuration for Document Conversion, see Setting up Document Conversion. **Note:** Because you need to use the Admin Console to create the connection, you should finish the installation and configuration steps in this checklist before you complete the Document Conversion node setup.
12	Configure a Binary Storage Provider	Required	With a supported web browser, navigate to `http://<hostname>/`, where `hostname` is the DNS-resolvable name of the server where you installed the Jive application on your primary web application node. Log in as the system administrator and use the instructions for setting up a binary storage provider, as described in Configuring binary storage provider.
13	Directory Server Integration	Optional	If you plan to populate your community with users synchronized from your LDAP or Active Directory implementation, refer to Setting up LDAP and Active Directory.
14	Post-installation Tasks	Required	Use the post-installation tasks, described in Post-installation tasks on page 41, to perform some one-time configurations for Jive, such as configuring the location for text extraction for search.

CLI installation tasks

Installation tasks include installing the RPM, using `jive setup`, and copying the configuration to other nodes in clustered environments.

Installing Jive package and starting up

The Jive command-line interface (CLI) provides a way to track down configurations and manage the platform.

Jive is compatible with a number of hardware configurations as well as network topologies. To understand the recommended deployment configuration for an on-premise installation, see Jive Enterprise architecture.

Requirements

To install Jive using the RPM, you need the following:

- At least six servers that meet the minimum specified hardware requirements described in System requirements.
- These servers should be running Linux. See the System requirements for supported versions.
- A bash shell to run the install commands.
- SSH access to the servers so you can copy the RPM there for installation.
- Ability to become the root user on the servers where the installation is performed, commonly via SSH, or less commonly through user interface access such as VNC.

> **Note:** If you want to run a non-standard installation, you could separate installation tasks into root user and non-root user tasks. You need a root user to complete the pre-installation requirements on CLI, and this root user would have to create a non-root user who can run the installation. For more information, see Pre-installation requirements as root on CLI.

- Make sure you have installed the packages described in RPM dependencies by operating system and you complete the pre-installation requirements on CLI described in Pre-installation requirements as root on CLI. They include copying the Jive and `pdswf` RPMs to the servers and modifying a few configuration files.

Install Jive package on all nodes

You need to install the Jive Linux package on the following nodes. You can install the RPM on these nodes in any order, but use the Installation overview on page 23 for the recommended order.

- Activity Engine node
- Search node
- Web Application nodes (2 or more, where node 1 is the primary Web Application node)

- Cache server
- Document Conversion node

Installation steps using Jive CLI

The following installation steps are the most common approach to installing the Jive platform.

1. From the command line, access the target host as root. For example, the following illustrates using the `ssh` command to access the server at targethost as the root user.

   ```
   joe@joesbox ~ $ ssh root@targetsystem
   root@targethost's password:
   Last login: Mon Feb 14 14:00:56 2011 from joesbox.example.com
   ```

2. If you haven't already copied the Jive application RPM to each server and application node in your Jive environment, then you should do it now.

   Here's an example using the Linux `scp` command to copy the package from a computer named `joesbox` to a target system at `targetsystem`:

   ```
   scp -v joe@joesbox:/Users/joe/jive.rpm root@targetsystem:/root
   ```

3. Install the Jive application RPM on all nodes by using an `rpm` command such as the following. The `U`, `h`, and `v` options are provided to indicate install or upgrade with hash indicators, and to be verbose during the installation.

   ```
   rpm -Uvh jive.rpm
   ```

   Your copy of the Jive RPM file — here, `jive.rpm` — will have a slightly different name.

4. When the installation finishes, the Jive CLI installer asks you to become a jive user and run `jive setup`, which you can do by using the following command. For more information, see Using jive setup on page 29.

   ```
   su - jive
   jive setup
   ```

5. Address any configuration issues and suggestions that `jive setup` provides. Re-run `jive setup` until you get the `setup ok` message.
6. Set up your Core application, Activity Engine, and Analytics (optional) databases, as described in Setting up new databases. If you are using a database whose driver is not included, ensure its driver is in the application's classpath.
7. After you have installed the Jive package and run `jive setup` on all of your nodes, you can enable the services as described in the next section.

Enabling services

1. Enable the services on all of the nodes as a jive user.

   Note: You can run `jive enable --help` to see what services are available to enable, or `jive status -v` to see all of the services and whether they're enabled or disabled.

On this node	Run this command as jive user
Activity Engine	`jive enable eae`
Search	`jive enable search`

Web application nodes	`jive enable webapp`
	`jive enable httpd`

Cache servers and cluster nodes	`jive enable cache`
Document Conversion	`jive enable docconverter`

2. After enabling the correct services, re-run `jive setup`. For more information, see Using jive setup on page 29. Do this so the Jive CLI installer can detect any service-specific startup properties that need to be set:

   ```
   jive setup
   ```

3. Once you get the `setup ok` message on your node, you can start the enabled services on Jive.

   ```
   jive start
   ```

4. With a supported web browser, go to `http://<hostname>/`, where `hostname` is the DNS-resolvable name of the server where you installed the Jive application on your primary web application node. There, you are prompted to finish configuring the Jive application via the Admin Console Setup wizard, as described in Configuring application with the Setup wizard on page 51. If you plan to populate your community with users synchronized from your LDAP implementation, the setup screens are included in this wizard, as described in Setting up LDAP and Active Directory.
5. See the post-installation tasks, described in Post-installation tasks on page 41, for your next steps.

Using `jive setup`

Use `jive setup` after you install the RPM. It provides system requirements and suggestions for configuring your Jive instance.

As you go through the `jive setup`, it provides suggestions for your configuration. Use the command-line output to update your configurations. After you make changes, run `jive setup` until you get the `setup ok` message.

Note: Running `jive start` automatically runs `jive setup` before starting jive unless you disable it as shown in the following section.

Disabling `jive setup`

You can skip jive setup by doing one of the following:

- Tell the start process to not use setup, and only start Jive:

  ```
  jive start --skip-setup
  ```

- Set the `main.skip_setup` startup property to true:

  ```
  jive set main.skip_setup true
  ```

Copying configurations to other nodes

After running the `jive setup` and configuring all of the overrides, you can move the overrides configuration file to another node that requires the same setup.

After going through `Jive setup`, the Jive CLI updates the `jive.properties` file, which is where it saves overrides to the startup properties. You can copy the `jive.properties` file to another node that needs the same setup.

For example, after configuring your first web application node, you can copy the *Jive Installation Directory*`/etc/jive.properties` file to the second node. You also need to manually copy the encryption keys in the `/usr/local/jive/applications/`*app_name*`/home/crypto` directory to the new node. After that, you can just run `jive start` to start the node.

Troubleshooting installation

The Jive installation uses Linux RPM, a widely tested and used application that is very unlikely to fail. However, if you run into trouble during installation, you can delete and start over.

Note: You can find the installation log files on the target computer at `/usr/local/jive/var/logs`.

Unsatisfied dependencies

The Jive application RPM depends on the presence of several low-level system packages that are common to nearly all configurations of Jive’s supported Linux distributions. Also, the Jive application RPM depends on three high-level packages. If any of these packages (system or high-level) is not present, the RPM subsystem warns you, then refuse to install. When you see these warnings, install the missing packages using RPM, then install Jive as described in the instructions.

Unsatisfied dependencies appear as an error when attempting to install the Jive application:

```
[root@targethost ~]# ls -l
total 202068
-rw-r--r-- 1 root root 206701420 Jan 20 16:03 jive_sbs-5.0.0-78310.i386.rpm
-rwxr-xr-x 1 root root      1347 Oct  7 16:14 updateDNS.sh
[root@targethost ~]# rpm -ivh jive_sbs-5.0.0-78310.i386.rpm
error: Failed dependencies:
```

```
     bash >= 3.2 is needed by jive_sbs-5.0.0-78310.i386
     sysstat >= 7 is needed by jive_sbs-5.0.0-78310.i386
```

Depending on the host configuration, it may be possible to install the dependencies directly using system tools. For example, in RedHat Enterprise Linux, the `yum` command can install dependencies via network repositories. The following demonstrates how to install the dependencies shown in the error above.

```
[root@targethost ~]# yum install bash-3.2 sysstat
Loading "installonly" plugin
Setting up Install Process
Setting up repositories
extras                    100% |=========================| 1.1 kB    00:00
updates                                                          951 B 00:00
base                      100% |=========================| 1.1 kB    00:00
addons                    100% |=========================|  951 B    00:00
Reading repository metadata in from local files
primary.xml.gz            100% |=========================|  90 kB    00:00
################################################## 295/295
primary.xml.gz                                                  369 kB 00:03
################################################## 796/796
primary.xml.gz            100% |=========================| 853 kB    00:01
################################################## 2458/2458
Parsing package install arguments
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Downloading header for sysstat to pack into transaction set.
sysstat-7.0.2-1.el5.i386. 100% |=========================|  15 kB    00:00
---> Package sysstat.i386 0:7.0.2-1.el5 set to be updated
---> Downloading header for bash to pack into transaction set.
bash-3.2-21.el5.i386.rpm  100% |=========================|  55 kB    00:00
---> Package bash.i386 0:3.2-21.el5 set to be updated
--> Running transaction check
Dependencies Resolved

=============================================================================
 Package                 Arch       Version          Repository        Size
=============================================================================
Installing:
 sysstat                 i386       7.0.2-1.el5      base              168 k
Updating:
 bash                    i386       3.2-21.el5       base              1.9 M
Transaction Summary
=============================================================================
Install      1 Packages
Update       1 Packages
Remove       0 Packages
Total download size: 2.0 M
Is this ok [y/N]: y
Downloading Packages:
(1/2): sysstat-7.0.2-1.el 100% |=========================| 168 kB    00:00
(2/2): bash-3.2-21.el5.i3 100% |=========================| 1.9 MB    00:02
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating  : bash                         ######################### [1/3]
  Installing: sysstat                      ######################### [2/3]
  Cleanup   : bash                         ######################### [3/3]
Installed: sysstat.i386 0:7.0.2-1.el5
Updated: bash.i386 0:3.2-21.el5
Complete!
```

After dependencies have been resolved, the package should install normally.

Insufficient system memory

The Jive platform requires a minimum of 3GB of RAM to operate effectively for an enterprise environment. If sufficient memory is not available on the target installation system, the installer provides a warning at installation time similar to the example below.

```
[root@targethost ~]# rpm -ivh jive_sbs-5.0.0-78310.i386.rpm
Preparing...                ########################################### [100%]
   1:jive_sbs               ########################################### [100%]
Writing installation version.
Wrote installation version.
Executing Jive post-install configuration.
Creating jive group jive.
Creating jive system user jive.
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
Marking all upgrades as complete.
WARNING: this host does not have sufficient RAM to run a production Jive system.
A minimum of 3GB is required to host the application and HTTPD servers. 4GB is
required to
run a locally hosted database.
Starting Jive System Daemon.
Performing Jive system configurations.
Disabling CPU frequency stepping.
. . .
Jive post-install configuration complete.
```

In the above example, note the message “WARNING: this host does not have sufficient RAM to run a production system. A minimum of 3GB is required to host the application and HTTPD servers. 4GB is required to run a locally hosted database.”

Despite this warning, the package does install correctly; however, further errors are noted on the output line: "Failed to start application sbs. See log file at '/usr/local/jive/var/logs/sbs.out'." The contents of this log file indicate:

```
[root@targethost ~]# cat /usr/local/jive/var/logs/sbs.out
SCRIPT_DIR=/usr/local/jive/applications/sbs/bin
JIVE_BASE=/usr/local/jive/applications/sbs

Creating temp directory at /usr/local/jive/var/work/sbs.
Starting application sbs
Error occurred during initialization of VM
Could not reserve enough space for object heap
```

Starting over

In the unlikely event that something goes wrong during installation and you want to start over, you can uninstall. When uninstalling, you don't specify the RPM filename, as you did when installing. Instead, provide the logical name by which the RPM now knows the application: `jive_sbs`. Here's an example using `rpm -e` for uninstalling:

```
rpm -e jive_sbs
```

If you want to be sure you've removed all remnants of the installation, delete the destination directory created by the RPM with:

```
rm -rf /usr/local/jive
```

Changing root context (optional)

Jive Apache uses the context element of a web application to delegate a URL to your Jive web application. You can set the context to something other than root if required.

By default, Jive installs and configures using the root context. For example, `http://yourcommunity.com`. This is the recommended configuration and default installation behavior, but you can also specify a non-root context such as `http://yourcommunity.com/engage`. If you've already installed Jive, see Changing root context on existing installation on page 33 to change the context to something other than root.

Important: Never use `community` as your non-root context because the application already uses this string.

To change the root context on a new install:

1. Add the non-root context to the startup property by using the following command, where `engage` is an example of adding new context. Setting this property makes the Jive web application available at `http://yourcommunity.com/engage` instead of `http://yourcommunity.com`.

   ```
   jive set webapp.context engage
   ```

2. Restart the Web Application node by using the following command.

   ```
   jive restart webapp
   ```

3. Restart the Jive Apache service by using the following command.

   ```
   jive restart httpd
   ```

Changing root context on existing installation

You can use these instructions if you've already installed Jive, and you need to change the context to something other than root.

To change the root context on an existing installation:

1. Add the non-root context to the startup property by using the following command, where `engage` is an example of adding new context. Setting this property makes the Jive web application available at `http://yourcommunity.com/engage` instead of `http://yourcommunity.com`.

   ```
   jive set webapp.context engage
   ```

2. Change the jiveURL system property to reflect the new context for the web application at **Admin Console: System** > **Management** > **System Properties** .

   For more information on the jiveURL, see Changing community URL.

3. Restart the Web Application node by using the following command.

   ```
   jive restart webapp
   ```

4. Restart the Jive Apache service by using the following command.

   ```
   jive restart httpd
   ```

Alternate installation scenarios with Jive CLI

You can use the Jive CLI to customize your Jive installation, so it fits your needs.

If the default method of installation doesn't work for you, choose a scenario that makes sense for your environment.

Installing Jive without root access

Create a new user who performs both the RPM install and runs the platform. You need to create this user in advance.

You need to use the following procedure on each node you're installing Jive.

Note: Installing without root access, does not install an init script or a logrotate cron task. To perform these tasks, you need root access to run `sbin/install_init_script_and_logrotate.sh`, or equivalent.

1. Someone with root access needs to review Pre-installation requirements as root on CLI and complete the required tasks.
2. Create a group for your user. We use `myjivegroup` in the following example.

   ```
   groupadd myjivegroup
   ```

3. Create the user and add it to the group you created in the previous step.

   ```
   useradd -g myjivegroup -c "My Jive Community Service Account" -d /opt/apps/myjive
    -m -s /bin/bash myjiveuser
   ```

   Note:

   - Make sure this user has `/bin/bash` as shell.
   - You must install Jive in the user's home directory. In this example, the installation directory is `/opt/apps/myjive`.

4. Initialize an RPM database as the new user in that user's home directory.

   ```
   su - myjiveuser
   mkdir rpmdb
   rpm --initdb --dbpath /opt/apps/myjive/rpmdb
   ```

5. After downloading the Jive RPM and copying it to the node where you're installing Jive, you need to verify that the dependencies are met, as described in RPM dependencies by operating system.

   ```
   rpm -Vp --nofiles path/to/jive.rpm
   ```

6. Install the RPM as your new user.

   ```
   JIVE_USER=myjiveuser JIVE_GROUP=myjivegroup JIVE_HOME=/opt/apps/myjive rpm
   --dbpath /opt/apps/myjive/rpmdb -Uvh --prefix=/opt/apps/myjive --nodeps
   path/to/jive.rpm ]
   ```

7. For the docconverter node, install the `pdf2swf` RPM.

   ```
   JIVE_USER=myjiveuser JIVE_GROUP=myjivegroup JIVE_HOME=/opt/apps/myjive rpm
   --dbpath /opt/apps/myjive/rpmdb -Uvh --prefix=/opt/apps/myjive --replacefiles
   --nodeps path/to/pdf2swf.rpm
   ```

8. Source the .bash_profile that was just installed, or log out and back in again as your new user. For example, `myjiveuser`.
9. Run `jive setup`, which you can do by using the following command. For more information, see Using jive setup on page 29.

   ```
   jive setup
   ```

Installing in a different directory with a different user

Install the RPM normally as root, but with an installation user and group other than `jive`, or an installation directory other than `/usr/local/jive`, or both.

With root access, you can install Jive and create a non-default user to run the platform. You can skip Steps 1 on page 35 and 2 on page 35 if you don't need to create your user ahead of time.

Note: If you are not creating a user in advance, you should specify a user and group during the RPM installation using the JIVE_USER and JIVE_GROUP environment variables. These variables default to `jive`. For an example of how to set these variables, see Step 3 on page 36.

To install and set up Jive:

1. Create a group for your user. We use `myjivegroup` in the following example.

   ```
   groupadd myjivegroup
   ```

2. Create the user and add it to the group you created in the previous step.

   ```
   useradd -g myjivegroup -c "My Jive Community Service Account" -d /opt/apps/myjive
    -m -s /bin/bash myjiveuser
   ```

   Note:

   - Make sure this user has `/bin/bash` as shell.
   - You must install Jive in the user's home directory. In this example, the installation directory is `/opt/apps/myjive`.

3. Log in as the root user to perform an RPM installation, so that the Jive platform is not runnable by the new user and installed in the new user's home directory In the following example, we use `myjiveuser`, `myjivegroup`, and `/opt/apps/myjive` as the user, group and installation directory.

   ```
   JIVE_USER=myjiveuser JIVE_GROUP=myjivegroup JIVE_HOME=/opt/apps/myjive rpm -Uvh
    --prefix=/opt/apps/myjive path/to/jive.rpm
   ```

Parameters	Description

`JIVE_USER=myjiveuser`
`JIVE_GROUP=myjivegroup`
	Uses these parameters to specify the non-default user that will run the platform. If you don't use these parameters, the values default to `jive` and `jive`. The RPM installer creates this user and group if it does not exist.

`JIVE_HOME=/opt/apps/myjive`
`--prefix=/opt/apps/myjive`
	Use `JIVE_HOME` and the `--prefix` flag to install Jive in an alternate location. Installing the RPM without using these parameters installs in the default location, which is `/usr/local/jive`.
	Note: You can change the installation directory even if you are using the default `jive` user.

4. For the DocConverter node, log in as the root user to install `pdf2swf` RPM.

   ```
   JIVE_USER=myjiveuser JIVE_GROUP=myjivegroup JIVE_HOME=/opt/apps/myjive rpm -Uvh
    --prefix=/opt/apps/myjive --replacefiles path/to/pdf2swf.rpm
   ```

5. When the installation finishes, become the user you created during the install and run `jive setup`, which you can do by using the following command. For more information, see Using jive setup on page 29.

   ```
   su - myjiveuser
   jive setup
   ```

Running Jive on port 80 without SSL

Although not recommended in production, you can run Jive on port 80 without using SSL if you configure webapp proxy settings and you redirect ports using iptables.

To honor a number of customer requests, we removed the requirement of being a privileged (root) user to install Jive. As a result, Jive now runs on port 8080 because Jive is now installed by non-privileged (`jive`) user and that user cannot use port 80.

Beginning with Jive 7, we insist you run Jive with the more secure SSL. If you want to change this, you now need to make a few Linux configurations to allow access to your site using port 80 without SSL.

To set up a Jive instance without SSL that listens on port 80:

1. Configure the webapp proxy settings using the following commands:

   ```
   jive set webapp.http_proxy_name <your_hostname_here>
   jive set webapp.http_proxy_port 80
   jive set webapp.http_proxy_scheme http
   ```

2. Redirect ports using iptables by using the following commands as root user:

   ```
   iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j REDIRECT --to-port
   8080
   iptables -A OUTPUT -t nat -p tcp -d 127.0.0.1 --dport 80 -j REDIRECT --to-ports
    8080
   iptables -A OUTPUT -t nat -p tcp -d $(host `hostname` | awk '{print $NF}')
   --dport 80 -j REDIRECT --to-ports 8080
   ```

3. If you want these settings to persist on reboot use the following commands, also as root user:

   ```
   mv /etc/sysconfig/iptables /etc/sysconfig/iptables-old && iptables-save >
   /etc/sysconfig/iptables
   chkconfig iptables on
   ```

Setting up Document Conversion

Some documents — including PDFs and some of the Office documents — are supported in a preview view in Jive. To convert content from its native format into a form that can be previewed without altering the original document, you need the Document Conversion module. This module you deploy on a server that is separate from your core Jive production instances.

Jive gives users the ability to upload Office and Adobe PDF documents to the community for easy content sharing and collaboration. The Document Conversion service converts uploaded documents to a standard PDF format and then converts them again to Adobe Flash (.swf files) so that they can then be viewed in a web browser without needing to open the document's native software application.

We support converting the following file types:

- doc
- ppt
- docx
- pptx
- xls
- xlsx
- pdf

Note: For more information about managing conversion attempts and reconverting documents if necessary, see Managing Document Conversion.

Here is an overview of the steps you perform to set up Document Conversion:

1. Set up a production instance of the Jive application. You should devote one node in your installation to document conversion. For more information, see Installing Jive package and starting up on page 27.
2. Install the Jive platform RPM on your conversion node machine, as described in Setting up conversion machine on page 41. Then disable the services not related to document conversion. Download and install the correct RPM for the PDF2SWF utility on the conversion node machine. You can find the RPMs at https://static.jiveon.com/docconverter.

   Note: If you receive the following error when installing either the `jive_sbs` rpm or `jive_pdf2swf`, use the `--replacefiles` flag on the document conversion node.

   ```
   file /usr/local/jive/bin/pdf2swf from install of rpm_name conflicts with file
   from package other_rpm_name the flag
   ```

3. Enable the Document Conversion service by using the following commands:

   ```
   jive enable docconverter
   jive start
   ```

4. On the application node, configure the application to communicate with the conversion machines, as described in Configuring Document Conversion node connection.
5. If you want to set up secure communication to the conversion machine, see Setting up SSL for Document Conversion.

Setting up SSL for Document Conversion

If you have an SSL certificate, you can set up secure communication by editing the `docconverter/conf/server.xml` file and specifying the new secure URL in your Document Conversion Settings.

Before you can set up secure communication with your Document Conversion server, you need to acquire an SSL certificate.

To add an SSL certificate to your instance:

1. Edit the `/usr/local/jive/services/docconverter/conf/server.xml` file and add a connector to listen on port `8443`.

   For example:

   ```
   <Connector port="8443" maxThreads="200" scheme="https" secure="true"
   SSLEnabled="true" SSLCertificateFile="/usr/local/jive/services/docconverter/home/jive.crt" SSLCertificateKeyFile="/usr/local/jive/services/docconverter/home/jive.key" clientAuth="optional" />
   ```

   where `SSLCertificateFile` is the certificate file and `SSLCertificateKeyFile` is the key file.

For more information on setting up Tomcat and https, see Apach documentation at http://tomcat.apache.org/tomcat-8.0-doc/ssl-howto.html.

2. Make sure the SSL engine is on.

   For example:

   ```
   <Listener className="org.apache.catalina.core.AprLifecycleListener"
   SSLEngine="on"/>
   ```

3. Restart the document conversion service by running the following command as the jive user:

   ```
   jive restart docconverter
   ```

4. Go to **System** > **Settings** > **Document Conversion Settings** and edit **Conversion Service Settings** to specify the new secure URL and port.

   For example:

   ```
   https://conversion-node:8443/conversion/v1
   ```

5. Verify that you can run all conversion tests successfully.

Configuring Document Conversion node connection

You use the Document Conversion Settings page to configure the node that hosts the core Jive application, so the main application knows how to communicate with any conversion machines you've set up. Before you use the settings on this page, you should already have set up a document conversion node. For an overview of Document Conversion setup, see Setting up Document Conversion.

Fastpath: > **Admin Console: System** > **Settings** > **Document Conversion**

The Document Conversion Settings page is pre-populated with the default values for most installations. Use the following steps to ensure your setup is correct.

To enable document conversion:

1. Under **Document Conversion**, select **Enabled** to enable this feature.
2. Click **Add a Node** to start configuring a conversion machine.
3. Enter the IP address or hostname for the conversion machine.
4. Switch the way you access the conversion service URL from http to https by setting this in the **Conversion Service Settings** field.
5. If you want to exclude some file types from conversion, enter a comma-separated list in **Disabled Extensions**. For example, if you don't want to convert Excel files, type `xls, xlsx`.

Troubleshooting Document Conversion machine setup

If your document conversion tests are failing, try investigating the areas described here to resolve the problem.

- Check that port `19003` is open between the application node and the conversion node by executing a telnet command. For example, run `telnet 10.61.32.156 19003`.
- Check the log for the DocConverter service for startup exceptions. You can find the log at `/usr/local/jive/var/logs/docconverter.out`.
- If the Office to PDF test is failing on the Document Conversion Setup page, execute `jive status -v` to verify that DocConverter has open ports `8820`, `8821`, `8822`, `8823`, `8824`.

Setting up cache server

Use the same Linux package that you used to install the application servers to install a separate cache server that application servers can use in a cluster.

If your installation uses a single application node, the installation does not enable the cache services; instead, the installation use the local cache installed with the application server. When you have a multi-node configuration, use the following steps to set up a cache service in the cluster.

To install a cache server:

1. Install the RPM as described in Installing Jive package and starting up on page 27.
2. Because the cache server machine's only function is operating as a cache server, only enable the cache service. To do this, type the following as jive user:

   ```
   jive enable cache
   ```

   Note: If you're logged in as root, you can use `su - jive` to become the jive user.

3. Configure the cache server with its address.
4. Register and start the caching service by using the following commands.

   ```
   jive set cache.hostnames hostname
   jive start
   ```

   For more information on setting up multiple cache servers and high-availability, see Configuring Cache servers for high-availability.

The cache service writes several log files to `$JIVE_HOME/var/logs/`. These are:

- `cache-service.out`: Cache startup messages, output from the cache processes, showing start flags, restarts, and general errors.
- `cache-gc.log`: Output from garbage collection of the cache process.

5. If you haven't already, set up your application cluster to use the cache server address. You can find this in the Admin Console, at **System** > **Settings** > **Caches** .

Setting up conversion machine

The document conversion machine should be a dedicated machine that isn't used for any other Jive services.

After you install Jive on the conversion machine, run `jive enable docconverter` as the jive user and disable all the services other than `jive-docconverter`.

For more information about installing Jive, see Installing Jive package and starting up on page 27.

Post-installation tasks

Here you can find sample configurations and script examples common to long-term operation of a Jive installation.

Using commands to work with your managed instance

Jive includes several command-line tools you can use to perform maintenance tasks with your managed instance. With these tools, you can, for example, start and stop the application, upgrade the application, and collect information needed by Jive support. For the list of available commands, see Application management command reference.

Enabling SSL encryption

Enabling encryption of HTTP traffic requires several steps on a platform-managed host: you need to enable this for your required load balancer. For more information, see Configuring SSL on load balancer.

Setting up Document Conversion node

Some documents — including PDFs and those from Microsoft Office — are supported in a preview view in Jive. If you want to convert content from its native format into a form that can be previewed without altering the original document, you need the Document Conversion module, which you need to deploy on a server that is separate from your core Jive production instances.

If you have purchased the Document Conversion module, you need to set up this module as described in Setting up Document Conversion.

Configuring text extraction location

For content search, all binary content uploaded to Jive, such as .doc, .ppt, .pdf, .txt, or .xls, goes through a process where Jive extracts the plain text from the documents so it can be used in the search index. By default, the output for this process is stored on the web app node in `/usr/local/jive/applications/sbs/home/search/search-text-extraction` (these are encrypted files). However, we strongly recommend you change this to an NFS-mounted directory on a different node. In clustered environments, the NFS directory must be shared by all web app nodes in the cluster.

To specify the new text extraction location, you need to set the `webapp.custom_jvm_args` by typing the following as root:

```
jive set webapp.custom_jvm_args " -Djive.text.extract.dir=/path/to/new/location"
```

Use `jive list set webapp.custom_jvm_args` to see if you have set this value. If you have you can add to the existing value.

Configuring services directory for On-Premise Search

When you install On-Premise Search onto a multi-homed machine and you use the default host value of `0.0.0.0.`, On-Premise Search may not choose the desired network interface. Therefore, if you are running On-Premise Search on a multi-homed machine, you need to explicitly configure which network interface you want to bind to by changing the host values in the `serviceDirectory.json` file. For more information, see Configuring Service Directory for On-Premise Search.

Setting up video with firewall

If you want to use the Video Upload feature and your community is behind a firewall, or you restrict end user network access, you could use IP whitelisting rules to allow network communication to Twistage, where we host videos. For more information, see Setting up your firewall for video.

Installing on cluster

Production configuration requires you to set up a clustered environment.

Before You Begin

Before you set up the nodes in a cluster, you should have already configured a cache server, as described in Setting up cache server on page 40. The cluster requires the presence of a cache server to cache data that should be available to all nodes in the cluster. If your cache server isn't configured and running, you won't be able to set up the cluster.

Note: Your license determines whether or not clustering is enabled and how many nodes are supported. To check on the number of clustered servers your license allows, see the license information in the Admin Console.

Topology

The nodes in a cluster need to be installed on the same subnet, and preferably on the same switch. You cannot install nodes in a cluster across a WAN.

Upgrading

Important: If you're upgrading and copying the home directory (such as `/usr/local/jive/applications/<instance_name>/home`) from the older installation, you must preserve the `node.id` file and the crypto directory from the home directory before starting the server. The value stored in this file must be unique among the cluster nodes; that is, each node in a cluster should have a unique value in the `node.id` file. You must preserve the `node.id` file because it plays a role in storing encrypted information in the cluster. If that file is lost, you lose access to the encrypted data.

If you are deploying a new cluster, it is permissible to copy the contents of the home directory from the first node (where you set up clustering) to subsequent nodes — except for the `node.id` file. Do not copy the `node.id` file to subsequent nodes. If the `node.id` file does not exist, the application generates a new file on startup.

The cache server must be cleared and restarted before the upgraded application server nodes are started and try to talk to the cache.

If you're upgrading a plugin, clear the cache for that plugin and shut down the cache server first.

Starting new cluster

Always wait for the first node in the cluster to be up and running with clustering enabled before you start other cluster nodes. Waiting for a minute or more between starting each node ensures the nodes are not in competition. As the senior member, the first node you start has a unique role in the cluster. For a clustering overview, see Clustering in Jive.

Clocks

The clocks on all machines must be synchronized for caching to work correctly. For more information, see Managing in-memory cache servers. Also, if you're running in a virtualized environment, you must have VMware tools installed to counteract clock drift.

Cluster node communication

- Do not put a firewall between your cache servers and your Jive application servers. If you do so, caching does not work. A firewall is unnecessary because your application servers are not sending untrusted communications to the cache servers, or vice versa. There should be nothing that might slow down communication between the application servers and the cache servers.
- All ports between the cache and web application servers must be open.
- Port 6650 should be blocked to *external* access (but not between the cluster nodes!) so that any access outside of the data center is disallowed. This is required to prevent operations allowed by JMX from being executed on the cache server.

Overview of cluster installation

1. Familiarize yourself with the system requirements, important information about software, hardware, and network requirements and recommendations, described in System requirements.
2. Provision a database server.
3. If you're going to use a separate server for binary storage, configure a binary storage provider, as described in Configuring binary storage provider.
4. If your community will use the document conversion feature, configure Document Conversion, as described in Setting up Document Conversion.
5. Install a cache server on a separate server, as described in Setting up cache server on page 40.
6. Install and configure the application on the first node in your cluster, as described in Setting up cluster.
7. Install and configure the application on the subsequent nodes in your cluster.

Installing on cluster

Important: If, as part of your new installation, you're setting up one node as a template, then copying the home directory (such as `/usr/local/jive/applications/`*`your_instance_name`*`/home`) to other nodes in the cluster, you must **remove** the `node.id` file and the crypto directory from the home directory before starting the server. The application will correctly populate them.

1. Use the Jive application package to set up a cache server on a separate machine. For more information, see Setting Up a Cache Server. Note the cache server address for use in setting up the application servers.
2. Before proceeding, make sure the cache server you set up is running. It must be running while you set up the application server nodes.
3. On each node in the cluster, install the application using the package (RPM on Linux), but don't run the Admin Console's Setup wizard.

   For more information on installing the application, see Installing Jive package and starting up on page 27.
4. Start the primary node and navigate to its instance with a web browser. In the setup screen provided, enter the address of the cache server you installed, then complete the Admin Console Setup wizard.
5. After you've finished with the Setup wizard, restart the node.
6. Copy the `jive.license` file, the `jive_startup.xml` file, the `search` and `crypto` folders from the `home` directory on the primary node to the `home` directory in each of the other nodes in the cluster.

   Thehomedirectoryistypicallyfoundhere:`/usr/local/jive/applications/`*`your_instance_name`*`/home`.
7. On each of the secondary nodes, **remove** the `node.id` file from the home directory. The application will correctly populate it on each node when they are started for the first time.

8. Start the application on each of the secondary nodes (`service jive-application start` followed by `service jive-httpd start`). Because they are connecting to the same database used by the primary server, each secondary node detects that clustering is enabled and picks up the configuration you set on the primary node.
9. Restart all the servers in the cluster to ensure that the address of each node in the cluster is known to all the other nodes.

Configuring cluster node (optional)

You can get information about any node in the cluster and make configuration changes to it from the Admin Console. You can enable or disable clustering from a node in the cluster and set the node's cluster address. This is unnecessary unless you want to ensure that the node has a particular TCP endpoint — IP address and port number.

By design, a node that is new to a cluster creates its own address and registers itself in the database as a member of the cluster. However, you need to manually copy the encryption keys from the `/usr/local/jive/applications/`*`app_name`*`/home/crypto` directory to the new node before you can start it successfully.

For information about how the clustering system works, see Clustering in Jive.

Fastpath: Admin Console: System > **Settings** > **Cluster**

Use the following settings on the Admin Console's Cluster page to get information and configure the cluster:

Setting	Description
Enable cluster	Select **Enabled** to enable this node for the cluster.
Cluster members	Lists the addresses of other nodes in this cluster. Select the **Remove** check box to have this node's address removed from the database.
Local Cluster Address	Displays this node's IP address and the port on which this node listens for others in the cluster. The IP address and port form the unique TCP endpoint for this node in the cluster.
Cluster Overview	Lists the nodes in this cluster.

Adding features with plugins

By purchasing optional plugins offered by Jive Software, you can enhance Jive with powerful features not included by default.

- For the latest versions of Jive-supported plugins refer to Jive Extensions Catalog on Worx.
- For information about using plugins refer to Customizing with plugins.
- For information about building plugins and adding new features to the application refer to Developing plugins.

Configuring delegated authentication

In delegated authentication, Jive delegates authentication to your user identity provider.

Note: For information on building delegated authentication support for your user identity provider, see Jive SBS Delegated Authentication on Worx.

Fastpath: **Admin Console: People** > **Settings** > **Delegated Authentication Settings**

Use the following high-level steps to understand the configuration process. The sections below provide more information on the settings themselves.

1. Select the **Enable Delegated Authentication** check box to reveal other configuration options.
2. Under **Services**, select the services for which you want authentication delegated.
3. Under **Options**, select optional features to go along with authentication.
4. Under **Service Location**, enter the URL at which your authentication web service can be found.
5. Test communication from the application to your web service.
   a. Enter a user name and password that provide access to the web service.
   b. Enter the IP address for this community.
   c. Click **Perform Test**.

Services

This section lists services provided by the application, and which can require authentication for access. Each of these represents a point of access for users. Select the services whose authentication requests should be delegated to the authentication provider you're describing in configuration here.

Web interface The application's browser-based user interface. This is what your users likely use most often.

Web services Access via SOAP- or REST-based web services.

RSS feeds Access via RSS or Atom calls, such as from a feed aggregator.

Options

These are optional actions you can have the delegated authentication feature perform.

Auto-create users Select this to have the application create internal user accounts for users it authenticates with your identity provider, but who aren't represented in the application's database yet.

Synchronize profile fields Select this to synchronize user profiles between the application's profile data and profile data stored by your identity provider.

Service location

The service address is the location at which to find your authentication web service.

Username A user name known to the user identity provider.

Password The password for the username provided.

Source IP An optional field if your authentication web service evaluates the IP address of the incoming request. For example, you might use this if you anticipate allowing access from only one IP address and you want to test that functionality here.

Configuring binary storage provider

By default, Jive stores binary content (attachments on blog posts, discussions, uploaded documents, images, profile pictures, and avatars) in the application database, but we recommend migrating binary content to a different binary storage provider. Storing binary content data in a location outside the database is efficient and scalable. For production Jive systems, we recommend using a shared file system for storing all binary content. Documents created in the content editor are not binary content and are stored in the application database.

Preparing to migrate to shared file system location

You need to migrate your binary storage provider if:

- You want to use a file system provider instead of the JDBC provider enabled by default.
- You want to use the provider you're currently using, but change how you're using it. For example, you might want to use a different file system location for the file system provider.

Before you migrate to a different provider, you should do the following:

- Back up your current storage provider's backing store.
- Run a successful migration on a backup of this system.

Note: Do not run the migration tool during a busy period.

- Make sure the file system location is always available to the application.
- Estimate the amount of space you need for shared storage. For more information, see Required external components and Sizing binary storage.

Note the following things as you go through the migration process:

- During the migration, any new binary content is stored in both the current and the new storage provider. Once the migration has completed successfully, the system switches over to the new provider.
- If any errors occur during migration, the system reverts back to the current storage provider settings to allow the system to continue to function normally. Errors are logged, as described in Using application logs. When you've addressed the errors, start over with migration.
- Note that binary content caching is disabled during migration. This might cause increased load on both the current storage provider as well as the application in general.
- Allow at least a couple of hours for binary storage migration unless you have a trivial amount of content, for example, if you're just starting a community.
- When configuring a file system provider, you specify a namespace. The namespace helps ensure that data is isolated for the application instance it belongs to. So, for example, if you had multiple instances (such as an internal and an external instance), you could use the same storage server while giving the storage provider for each instance a different namespace.

Migrating to shared file system for binary storage

Fastpath: **Admin Console: System** > **Settings** > **Storage Provider**

To migrate to a shared file system:

1. Prepare a single unit of shared NFS or block-level storage on your SAN/NAS, and mount it from each of your web application nodes using the same mount point on each node (for example, `/opt/jiveBinStore`).

For information on estimating the amount of shared storage you need, Sizing binary storage.

2. From the **Storage Provider** page in the Admin Console, click **Migrate to another Storage Provider**.
3. In **Registered Providers**, select **FileStorageProvider**.
4. Click **Continue**.
5. Enter a namespace that corresponds to binary data for this application instance.
6. Enter the path to the local directory that you mapped to a mountable location in Step 1 on page 48.
7. Click **Continue**.
8. Read the notes, review, and confirm that the settings shown are the values you specified.
9. To delete binary content from the current source storage after migration, select the **Delete binary content** check box.
10. Click **Start Migration** to begin the process.
11. When the migration is complete, restart the application.
12. Disable local file caching:
    a. Go to **System** > **Settings** > **Storage Provider** .
    b. Click **Edit** under **Caching**.
    c. Select **No** under **Cache Enable**.
    d. Click **Save**.

JDBC storage providers

You can also migrate content to JDBC storage providers (a DBMS).

To migrate to a different provider:

1. In the Admin Console, go to **System** > **Settings** > **Storage Provider** .
2. Click **Migrate to another storage provider**.
3. In **Registered Providers**, select **JdbcStorageProvider**.

About using databases for binary storage

When you use another database for binary content storage, you specify a JNDI name corresponding to the database. During migration to the new provider, the application creates the database tables needed.

In a cluster, the JNDI name must be the same on each node where you configure an instance to use the provider.

Backup and storage considerations

Here you can find considerations for reliable storage and backups for your system.

Storage reliability

Here you can find the considerations for configuring reliable storage.

You should mount the Jive system home directory (by default, this is `/usr/local/jive`) on redundant external storage (preferably SAN storage via redundant HBAs and SAN fabric). When redundant external storage is not available, the local system volume for your Jive installation directory should be mirrored across multiple physical disks to prevent the loss of a single disc as a single point of failure.

The total storage requirements for this directory vary from installation to installation. As a basic guide for capacity planning, consider the following:

Core binaries
The base installation requires 500 MB storage (200 MB on disk, an additional 300 MB needed during upgrades of the platform).

Total system traffic
The system writes all logs to `<your Jive installation directory>/var/logs`. While the system by default rotates log files to reduce disk space consumed, larger installations may require to retain log files for analysis over time (for example, HTTPD access logs). In a default installation, allocating 5 GB for log storage should provide enough space to grow.

Cache efficiency
For each application, local caches of binary content including attachments and images are maintained. The more space available to those caches, the more efficient the system is at serving binary requests and the smaller the strain on the backing RDBMS. As a capacity guideline, plan on roughly .25 the planned total binary (BLOB) storage in the RDBMS for efficient caching.

Search index size
Each node stores local copies of the system search index. Generally, plan for search indexes to be 1x the total database storage consumption (.5 for active indexes, . 5 for index rebuilds).

Local database backups
When using the Jive platform-managed database, the database is regularly backed up to `<your Jive installation directory>/var/data/backup/full` and database checkpoint segments backed up to `<your Jive installation directory>/var/data/backup/wal`. When an instance is using this database, approximately 35x the total database size is required in the `<your Jive installation directory>/var/data/backup` location with a default configuration. This number can be lowered by more aggressively removing full backup archives stored in `backup/full` and by more aggressively removing WAL segments after a full backup has been performed.

Storage monitoring

As with any system, disk consumption should be regularly monitored and alerts generated when the system approaches disk capacity.

Most disk consumption occurs in three areas:

- **The application instance home directory**: By default, the platform manages a single application instance located at `<your Jive installation directory>/applications/sbs` with a home directory of `sbs/home`.
- **Platform logs**: All platform log files are stored in `<your Jive installation directory>/var/logs`
- **Platform database**: If the local platform database is used, data files are stored by default in `<your Jive installation directory>/ var/data/postgres-9.x` and backups in `<your Jive installation directory>//var/data/backup`.

System backups

You should perform regular backups of your instances.

In addition to performing regular backups of reliable storage, as described in Storage reliability on page 50, you should perform the backups:

- All databases
- Binary storage
- The Jive installation directory (by default, this is `/usr/local/jive`)

The simplest solution is to back up the entire contents of the Jive installation directory. A more selective option is to back up only the `/applications` and `/etc` subdirectories. In either case, you should make backups in accordance with standard backup practices.

Before you upgrade Jive, you should make a full backup of the entire installation directory.

Configuring application with the Setup wizard

> *From the Setup wizard, you enter information about your license, database connection, user identity system, email system, and system admin credentials.*

After you've installed the application, the Setup wizard runs automatically. Here you can find an overview of the setup process.

You can rerun the Setup wizard if required, as described in Starting Setup wizard manually on page 57.

License Settings

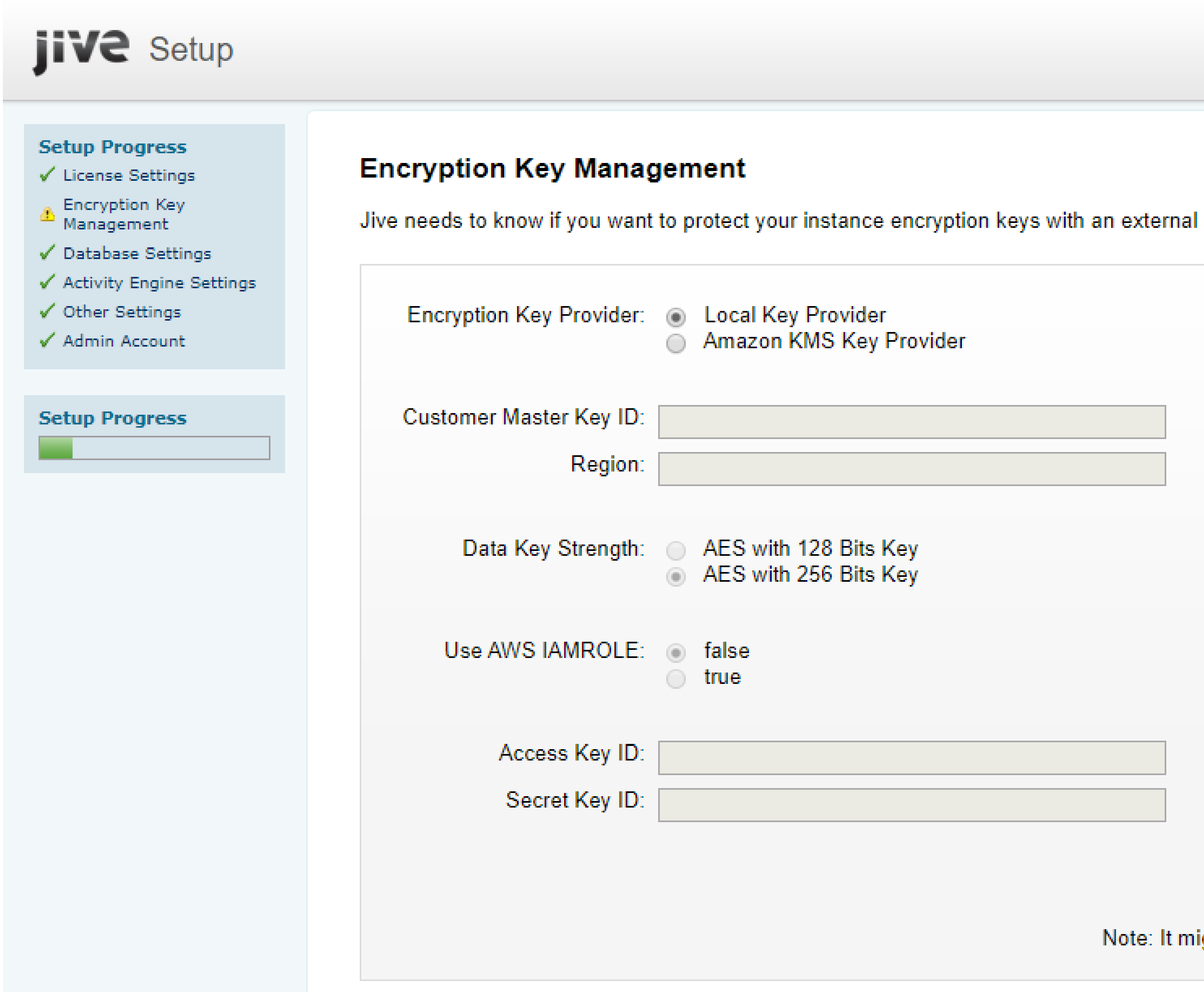
On the **License Settings** page, you enter the license key you purchased (you can retrieve your purchased keys by logging into `www.jivesoftware.com`) or click **Continue** to accept the local system license.

Next, on the **License** page, you select the type of deployment environment: production or development.

Encryption Key Management

All customer data is encrypted anywhere it is at rest. On the **Encryption Key Management** page, you can choose how you want to manage keys.

Figure 1: An Encryption Key Management page



Encryption Key Management

Jive needs to know if you want to protect your instance encryption keys with an external

Encryption Key Provider: ● Local Key Provider ○ Amazon KMS Key Provider

Customer Master Key ID:

Region:

Data Key Strength: ○ AES with 128 Bits Key ● AES with 256 Bits Key

Use AWS IAMROLE: ● false ○ true

Access Key ID:

Secret Key ID:

Note: It mi

If you choose **Local Key Provider**, Jive stores a master key used to encrypt and decrypt data on the server's file system. This is the default option.

Alternatively, you can choose to leverage an Amazon Web Services (AWS) account to use AWS Key Management Service (KMS) to encrypt and decrypt the secondary keys — **Amazon KMS Key Provider**. If you choose this option, you need to configure the following settings:

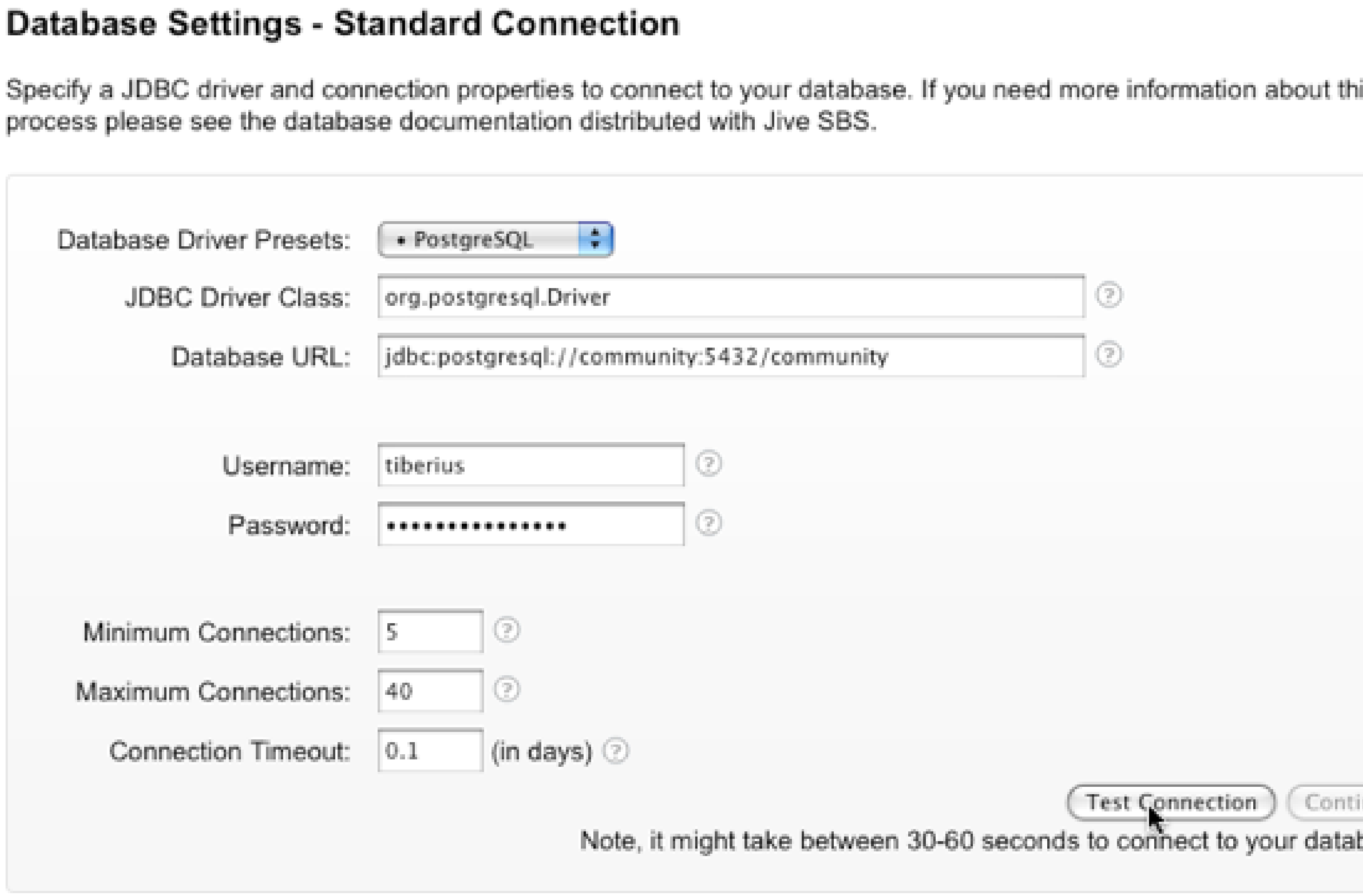
- **Customer Master Key ID**: The ID of the CMK previously generated on AWS KMS. It needs to be a symmetric key.
- **Region**: The AWS Region where the AWS KMS service is configured.
- **Data Key Strength**: AES encryption strength. You can choose to use 128 or 256-bits encryption. We recommend to use AES-256; this is the default option.
- **Use AWS IAMROLE**: If Jive is running on AWS EC2, you can configure an *IAM instance role* on the server and Jive will automatically pick it up. In this case, set this property to **true**. Otherwise, leave it as **false** and configure the IAM access keys.
- **Access Key ID** and **Secret Key ID**: The keys provided by AWS IAM to access AWS services programmatically. The IAM user requires read access to the appropriate AWS KMS Customer Master Key. These keys are used if you do not use a pre-configured IAM instance role.

After you specify the parameters, click **Test Connection with KMS** to ensure that details provided are appropriate and that the connection can be established successfully.

Database settings

On the **Database Settings** page, you specify how to connect to the application database. You can use the included local system database, or you can choose an external database. If you choose the standard database connection or JNDI data-source, you are prompted for required settings after you click **Continue**.

Figure 2: An example of a standard connection configuration



Database Settings - Standard Connection

Specify a JDBC driver and connection properties to connect to your database. If you need more information about thi process please see the database documentation distributed with Jive SBS.

Database Driver Presets: • PostgreSQL

JDBC Driver Class: org.postgresql.Driver

Database URL: jdbc:postgresql://community:5432/community

Username: tiberius

Password: ••••••••••••••••

Minimum Connections: 5

Maximum Connections: 40

Connection Timeout: 0.1 (in days)

Note, it might take between 30-60 seconds to connect to your datab

User settings

On the **User Settings** page, you specify the system that the application should use for the user and group data. By default, the Setup wizard uses the user and group database tables from the application database.

- If you choose **Default** and then click **Continue**, you go to the **User Profile Name Configuration** page. There, you are prompted to specify whether user profiles should provide combined or separate fields for a user to enter their name. For a new installation, we recommend that you specify separate fields. If you're upgrading from an instance that used one field, you should probably keep one field.
- Next, on the **Username Case Sensitivity Configuration** page, choose whether case sensitivity should be used to look up user names during account creation or login. For a new installation using the default user system, you should choose

case-insensitive lookups. Note that with this option, user name values must differ from one another in ways other than case alone.

Activity engine settings

On the **Activity Engine Settings** page, you specify the endpoints and database settings for the Activity Engine.

Other settings

You should specify defaults for feeds (such as RSS) and email sent and received by the application.

You can change these settings later in the Admin Console.

1. Set site basics, including its URL and name.

Setting	Description
Site URL	The URL that people can enter in the browser address bar to reach the community.
Space Name	The name that should be displayed as the name of the community, such as on the home page.

2. If this node is an application server node that's part of a cluster, use these settings to enable clustering for this node and to specify the address of the cache server that the cluster will use. For more information on clustering, see Clustering in Jive and In-memory caching.

Setting	Description
Clustering Enabled or Disabled	Select **Enabled** to indicate that this node is part of a cluster of application server nodes. If you enable clustering, you need to specify at least one cache server address.
Cache Server Addresses	For clustered installations, enter the domain name or IP address of at least one cache server.

3. Connect to your Search service. You have the option of either connecting to Jive's Cloud Search service or installing and connecting to an On-Premise Search service. Use these settings to connect to your Search service.

Setting	Options	Description
On-Premise Search	On-Premise Search Service Host	Enter the host name that Jive can use to connect to the Search service. For On-Premise Search, this is the host name for the node where you installed Search. The default is `localhost`. Refer to system requirements for information on Search node size.
	On-Premise Search Service Port	Enter the TCP port that Jive can use to connect to the Search service. The default is `30000`.
Cloud Search	Search Endpoint	For Cloud Search, select the public cloud search endpoint in your area. For connection information, see Search service on page 21.

4. The application can send and receive email. Configure these settings here.

a. Enter information to be used outgoing emails sent from the community, typically to users.

Setting	Description
Server Email Name	The name displayed in the From box in email sent from the community, such as to welcome new users.
Server Email Address	The email address for the sender in email sent from the community.

b. Set details for handling outgoing email. For more information on mail server configuration, see Configuring outgoing (SMTP) email.

Setting	Description
Email Host	Host name of an SMTP email server that the community will use to send email.
Email Port	Port to be used for sent email.

c. Set details for handling incoming email. For more information, see Configuring incoming email.

Setting	Description
Client Protocol	Select the protocol used by the incoming email server that the community will use when receiving email to post discussion replies.
Client Account Host	Host name of the incoming email server.
Client Account Port	Port of the incoming email server.
Username	Username for authenticating with the incoming email server.
Password	Password for authenticating with the incoming email server.

Administrator account

You should specify the system administrator's name, email, and password.

Note: The default user name and password is `admin`.

Restarting your application

After you finish with the Setup wizard, you are prompted to restart the application. You can do this by using the following commands from a command prompt as the jive user on the target computer:

```
[root@targethost ~]# sudo su - jiver
[1016][jive@targethost:~]$ jive restart webapp
```

When the application restarts, the following notification is displayed:

```
sbs restarted successfully.
```

Finishing up

After you've successfully finished with the Setup wizard, the application is ready to run. When you next log in to the user interface, you can continue setting it up. For example, you can create your user profile, update the interface theme, start creating simple space hierarchy. You can also send invitations to people to join the community. For more information about community setup, see Setting up community.

Starting Setup wizard manually

The Setup wizard runs the first time you access the application after installing it. Alternatively, you can run it manually if, for example, you need to use the tool to configure LDAP.

If you've already completed the setup process and need to use the tool again to configure LDAP, do the following:

To run the Setup wizard:

1. Stop the web application and httpd services by running `jive stop webapp` followed by `jive stop httpd`.
2. In the `/usr/local/jive/applications/sbs/home/jive_startup.xml` file, set the `<setup>` element to `false`.

   This resets the status to "setup has not been run".
3. Start the web application and httpd services by running `jive stop webapp` followed by `jive stop httpd`.
4. Open the Admin Console in your browser by typing in the URL.

   By default, using the built-in application server, this URL is: `http://<webapp_hostname>:8080/admin`.

The Setup wizard runs automatically.

Quick installation for Jive evaluation

Use this quick installation procedure to set up a small instance of Jive for evaluation and testing purposes only.

Note: The following steps assume you are installing Jive on a supported Linux operating system and you've never had Jive installed on the computer you're using for this evaluation. This procedure also assumes you are not running the Document Conversion feature or are tying any additional web application cluster or cache nodes to the environment. For the list of supported operating systems, see Supported operating systems.

1. Review the System requirements.

   Although this is an evaluation of Jive, you still need to install it on a supported Linux operating system and make sure it can communicate with remote resources. If you use a proxy server, make sure it's configured correctly to access the required external resources. For more information, see Supported operating systems, Preparing to connect to Jive Hosted services on page 19, and Connecting through proxy server on page 18.

2. Copy the Jive application RPM package to all nodes in the environment.

   Here's an example using the Linux `scp` command to copy the package from a computer named `joesbox` to a target system at `targetsystem`:

   ```
   scp -v joe@joesbox:/Users/joe/jive.rpm root@targetsystem:/root
   ```

3. Install the Jive RPM package on all nodes by using a command such as **`rpm -Uhv jive.rpm`**.

   Note that your copy of the Jive RPM package (called `jive.rpm` in this example) will have a slightly different name.

4. Configure Jive using the setup wizard by running `jive setup` on all of the nodes.

   Note that you must be the jive user to run `jive setup`. You may need to re-run `jive setup` until you get the `setup ok` message.

   Note: If you're logged in as root, you can use `su - jive` to become the jive user.

5. Install and configure the Activity Engine Database, Core Databases, and the optional Analytics Database.

For evaluation purposes, you can install the Activity Engine, Analytics, and Core databases together, but this is not supported in a production environment. The simplest way to set up a database on Linux is to perform a quick database setup for evaluation using a Postgres database, as described in Quick database setup for evaluations. If you are using a database whose driver is not included, ensure its driver is in the application's classpath.

6. Start the services. Note that you must be the jive user to run these commands:

```
jive enable eae
jive enable search
jive enable webapp
jive enable httpd
```

Note: If you're logged in as root, you can use `su - jive` to become the jive user.

7. Re-run `jive setup` on all of the nodes.

   This causes the installer to detect any service-specific startup properties that need to be set. Run `jive setup` until you get the `setup ok` message.

8. After you see the `setup ok` message, start the instance as follows:

```
jive start
```

9. From a web browser, open your new community and finish configuring it from the Admin Console Setup wizard.

   With a supported web browser, navigate to `http://<hostname>:8080`, where hostname is the DNS-resolvable name of the server where you installed the Jive application. For more information, see Configuring application with the Setup wizard on page 51.

10. Integrate your Directory Server.

    If you plan to populate your community with users synchronized from your LDAP or Active Directory implementation, see Setting up LDAP and Active Directory.