



OPERATIONS GUIDE

A guide for administrators of CircleCI Server installations on AWS and private infrastructure.

Docs Team

Version 2.17.2, 07-24-19

Overview	1
Build Environments	1
Architecture	2
CircleCI Server Container Architecture	5
Notes	5
key	5
Containers, Roles, Failure Modes and Startup Dependencies	5

Overview



CircleCI Server v2.17 uses the CircleCI 2.0 architecture.

CircleCI Server is a modern continuous integration and continuous delivery (CI/CD) platform installable inside your private cloud or data center. Refer to the [Changelog](#) for what's new in this CircleCI Server release.

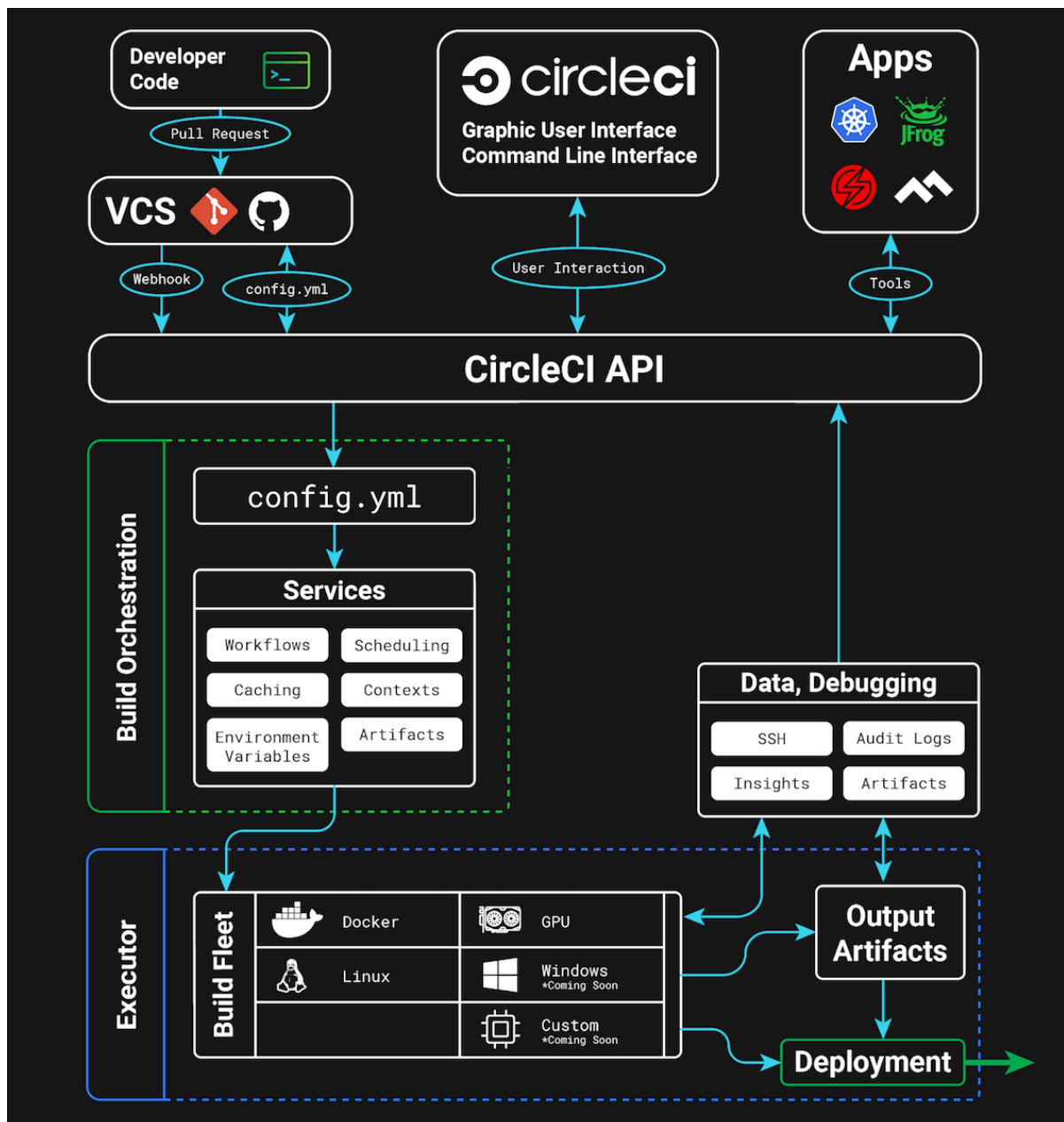


Figure 1. CircleCI Services Architecture

Build Environments

CircleCI 2.0 uses Nomad as the primary job scheduler. Refer to our [Nomad Introduction](#) to learn more about the job scheduler and how to perform basic client and cluster operations.

By default, CircleCI 2.0 Nomad clients automatically provision containers according to the image configured

for each job in your `.circleci/config.yml` file.

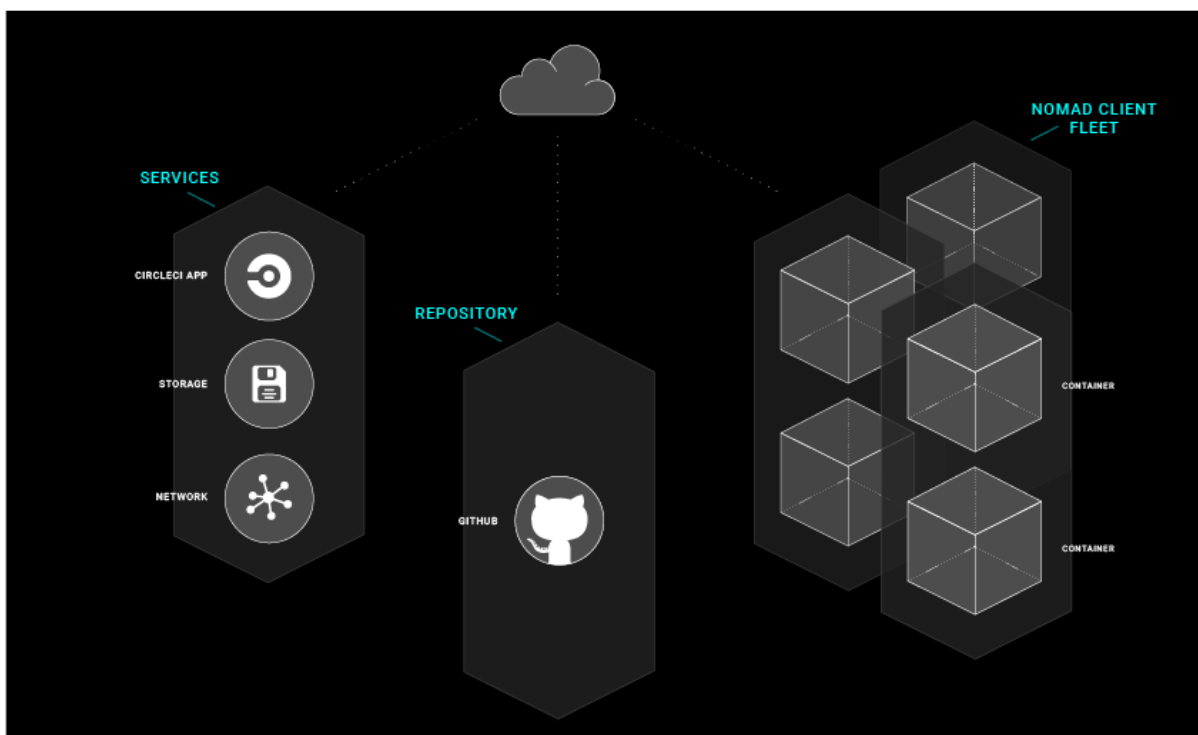
Architecture

Figure 1.1 illustrates CircleCI core components, build orchestration services, and executors. The CircleCI [API](#) is a full-featured RESTful API that allows you to access all information and trigger all actions in CircleCI.

Within the CircleCI UI is the Insights page, which acts as a dashboard showing the health of all repositories you are following including:

- median build time
- median queue time
- last build time
- success rate
- parallelism.

CircleCI consists of two primary components: Services and Nomad Clients. Any number of Nomad Clients execute your jobs and communicate back to the Services. All components must access GitHub or your hosted instance of GitHub Enterprise on the network, as illustrated in Figure 2.



Services Machine

The Services machine must not be restarted and may be backed up using VM snapshotting. If you must restart the Services machine, do so only as a last resort, because a restart will result in downtime. Refer to the [Disaster Recovery](#) chapter for instructions.

DNS resolution may point to the IP address of the Services machine. It is also possible to point to a load balancer, for example an ELB in AWS. The following table describes the ports used for traffic on the Service

machine:

Source	Ports	Use
End Users	80, 443, 4434	HTTP/HTTPS Traffic
Administrators	22	SSH
Administrators	8800	Admin Console
Builder Boxes	all traffic, all ports	Internal Communication
GitHub (Enterprise or .com)	80, 443	Incoming Webhooks

Nomad Clients

Nomad Clients run without storing state, enabling you to increase or decrease the number of containers as needed.

To ensure enough Nomad clients are running to handle all builds, track the queued builds and increase the number of Nomad Client machines as needed to balance the load. For more on tracking metrics see [Monitoring Your Installation](#).

Each machine reserves two vCPUs and 4GB of memory for coordinating builds. The remaining processors and memory create the containers. Larger machines are able to run more containers and are limited by the number of available cores after two are reserved for coordination.



The maximum machine size for a Nomad client is 128GB RAM/ 64 CPUs, contact your CircleCI account representative to request use of larger machines for Nomad Clients.

The following table describes the ports used on Nomad clients:

Source	Ports	Use
End Users	64535-65535	SSH into builds
Administrators	80 or 443	CCI API Access
Administrators	22	SSH
Services Machine	all traffic, all ports	Internal Comms
Nomad Clients (including itself)	all traffic, all ports	Internal Comms

GitHub

CircleCI uses GitHub or GitHub Enterprise credentials for authentication which, in turn, may use LDAP, SAML, or SSH for access. This means CircleCI will inherit the authentication supported by your central SSO infrastructure.



CircleCI does not support changing the URL or backend Github instance after it has been set up. The following table describes the ports used on machines running GitHub to communicate with the Services and Nomad Client instances.

Source	Ports	Use
Services	22	Git Access
Services	80, 443	API Access
Nomad Client	22	Git Access
Nomad Client	80, 443	API Access

CircleCI Server Container Architecture

This document outlines the containerized services that run on the Services machine within a CircleCI Server installation. This is provided both to give an overview of service operation, and to help with troubleshooting in the event of service outages. Supplementary notes and a key are provided below the following table.

Notes

- Database migrator services are listed here with a low failure severity as they only run at startup, however:



if migrator services are down at startup connected services will fail

- With a platinum support contract some services can be externalized (marked with * here) and managed to suit your requirements. Externalization provides higher data security and allows for redundancy to be built into your system.

key

Icon	Description
☑	Failure has a minor affect on production - no loss of data or functioning.
⚠	Failure might cause issues with some jobs, but no loss of data.
💣	Failure can cause loss of data, corruption of jobs/workflows, major loss of functionality.

Containers, Roles, Failure Modes and Startup Dependencies

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
api-service	Provides a GraphQL API that provides much of the data to render the web frontend.	Many parts of the UI (e.g. Contexts) will fail completely.	💣	postgres, frontend, contexts-service-migrator, contexts-service, vault-cci
audit-log-service	Persists audit log events to blob storage for long term storage.	Some events may not be recorded.	☑	postgres, frontend

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
<code>contexts-service</code>	Stores and provides encrypted contexts.	All builds using Contexts will fail.	⚠	<code>postgres</code> , <code>frontend</code> , <code>contexts-service-migrator</code> , <code>vault-cci</code>
<code>contexts-service-migrator</code>	Runs postgresql migrations for the <code>contexts-service</code>	Only runs at startup.	✅	<code>postgres</code> , <code>frontend</code>
<code>cron-service</code>	Triggers scheduled workflows.	Scheduled workflows will not run.	⚠	<code>postgres</code> , <code>frontend</code> , <code>cron-service-migrator</code>
<code>cron-service-migrator</code>	Runs postgresql migrations for the <code>cron-service</code> .	Only runs at startup.	✅	<code>postgres</code> , <code>frontend</code>
<code>domain-service</code>	Stores and provides information about our domain model.	Workflows will fail to start and some REST API calls may fail causing 500 errors in the CircleCI UI.	⬆	<code>postgres</code> , <code>frontend</code> , <code>domain-service-migrator</code>
<code>domain-service-migrator</code>	Runs postgresql migrations for the <code>domain-service</code> .	Only runs at startup.	✅	<code>postgres</code> , <code>frontend</code>
<code>exim</code>	Mail Transfer Agent (MTA) used to send all outbound SMTP.	No email notifications will be sent.	✅	None
<code>federation-service</code>	Stores user identities (LDAP)	If LDAP authentication is in use, all logins will fail and some REST API calls might fail.	⬆ only if LDAP in use	<code>postgres</code> , <code>frontend</code> , <code>federations-service-migrator</code>
<code>federation-service-migrator</code>	Runs postgresql migrations for the <code>federations-service</code> .	Only runs at statup.	✅	<code>postgres</code> , <code>frontend</code>

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
fileserved	File storage service used as a replacement for S3 when CircleCI Server is run outside of AWS. Not used if Server is configured to use S3. Stores step output logs, artifacts, test results, caches and workspaces.	If not using S3, builds will produce no output and some REST API calls might fail.	🔧 if not using S3	None
frontend	CircleCI web app and www-api proxy.	The UI and REST API will be unavailable and no jobs will be triggered by Github/Enterprise. Running builds will be OK but no updates will be seen.	⚠️	postgres
mongo*	Mongo data store.	Potential total data loss. All running builds will fail and the UI will not work.	🔧	mongodb-upgrader
nomad-metrics	Queries the nomad server for stats and sends them	Nomad metrics will be lost, but everything else	! [Mild] (images/circle-success.png) {width=20px}	None \
		to statsd. \	should run as normal. \	
	----- ----- ----- ----- ----- ----- ----- ----- ----- -----	output-processor / output-processing	Receives job output & status updates and writes	All running builds will either fail or be left in

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
![[Severe]](images/circle-failure.png){width=20px}	None \			them to MongoDB. Also provides an API to running
an unfixable, inconsistent state. There will also				
jobs to access caches, workspaces, store caches,	be data loss in terms of step output, test			
	workspaces, artifacts, & test results. \	results and artifacts. \		
----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----	permissions-service	Provides the CircleCI permissions interface. \	Workflows will fail to start and some REST API	![[Moderate]](images/circle-warning.png){width=20px}
postgres \				calls may fail, causing 500 errors in the UI. \
	frontend \			
		permissions-service-migrator \	----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----	permissions-service-migrator

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
Runs postgresql migrations for the	Only runs at startup. \	![Mild](images/circle-success.png){width=20px}	postgres \	
	permissions-service \			frontend \
----- ----- ----- ----- ----- ----- ----- ----- ----- -----	picard-dispatcher	Splits a job into tasks and sends them to	No jobs will be sent to Nomad, the run queue will	![Moderate](images/circle-warning.png){width=20px}
None \			schedulere to be run. \	increase in size but there should be no
meaningful loss of data. \			----- ----- ----- ----- ----- ----- ----- ----- ----- -----	postgres / postgres-script-enhance *
Basic postgresql with enhancements for creating	Potential total data loss. All running builds	![Severe](images/circle-failure.png){width=20px}	None \	
	required databases when containers are launched.\	will fail and the UI will not work. \		

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
----- ----- ----- ----- ----- ----- ----- ----- ----- -----	rabbitmq / rabbitmq- delayed *	Runs the RabbitMQ server. Most of our services	Potential total data loss. All running builds	![Severe](images/circle-failure.png){width=20px}
None \			use RabbitMQ for queueing. \	will fail and the UI will not work. \
			----- ----- ----- ----- ----- ----- ----- ----- ----- -----	outputRunningRedis / redis *
The Redis key/value store. \	Lose output from currently-running job steps. API	![Moderate](images/circle-warning.png){width=20px}	None \	
		calls out to github may also fail. \		

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
----- ----- ----- ----- ----- ----- ----- ----- ----- -----	schedul er er	Sends tasks to server-nomad to run. \	No jobs will be sent to Nomad, the run queue will	![Moderate](images/circle-warning.png){width=20px}
None \				increase in size but there should be no
meaningful loss of data. \			----- ----- ----- ----- ----- ----- ----- ----- ----- -----	mongodb-upgrader / server-mongo-upgrader
Used to run any mongo conversion/upgrade scripts	Not required to run all the time \	![Mild](images/circle-success.png){width=20px}	None \	
	during mongo version upgrade. \			

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
----- ----- ----- ----- ----- ----- ----- ----- ----- -----	nomad_server / server-nomad*	Nomad primary service. \	No 2.0 build jobs will run. \	![Severe](images/circle-failure.png){width=20px}
None \				
		----- ----- ----- ----- ----- ----- ----- ----- ----- -----	ready-agent / server-ready-agent	Called by Replicated to check whether other
Only required on startup. If unavailable on	![Mild](images/circle-success.png){width=20px}	None \		
containers are ready. \	startup the whole system will fail. \			----- ----- ----- ----- ----- ----- ----- ----- ----- -----
server-usage-stats	Sends the user count to the internal CircleCI “phone home”	CircleCI will not receive usage stats for your	![Mild](images/circle-success.png){width=20px}	None \

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
		endpoint. \	install but no affect on operation.	
	----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----	shutdown-hook-poller	Checks the frontend container for 1.0 Builder	1.0 Builder lifecycles will not be properly
![[Mild]](images/circle-success.png){width=20px}	None \			shutdown requests. If a request is found, the 1.0
managed, but jobs will continue to run. \				
Builder is shut down. \				----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
slanger	Provides real-time events to the CircleCI app. \	Live UI updates will stop but hard refreshes will	![[Mild]](images/circle-success.png){width=20px}	None \
			still work. \	

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
	----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----	telegraf	This is the statsd forwarding agent that our	Metrics will stop working but jobs will continue
![[Mild]](images/circle-success.png){width=20px}	None \			local services write to and can be configured
to run. \				
to forward to an external metrics service. \				----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
tutum/logrotate	Used to manage log rotations for all containers	If this stays down for a long period the Services	![[Moderate]](images/circle-warning.png){width=20px}	None \
		on the services machine. \	machine disk will eventually run out of space and	
				other services will fail. \

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
		----- ----- ----- ----- ----- ----- ----- ----- ----- -----	test-results	Parses test result files and stores data. \
There will be no test failure or timing data for	![Mild](images/circle-success.png){width=20px}	None \		
	jobs, but this will be back-filled once the			
		service is restarted. \		
----- ----- ----- ----- ----- ----- ----- ----- ----- -----	contexts-vault / vault-cci *	Instance of Hashicorp's Vault – an encryption	contexts-service will stop working, and all	![Moderate](images/circle-warning.png){width=20px}
None \			service that provides key-management, secure	jobs that use contexts-service will fail. \
				storage, and other encryption related services.
Used to handle the encryption and key store for				

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
	the contexts-service .\			
----- ----- ----- ----- ----- ----- ----- ----- ----- -----	vm-gc	Periodically check for stale machine and remote Docker instances	Old vm-service instances might not be destroyed	![Mild](images/circle-success.png){width=20px}
vm-service-db-migrator \			and request that vm-service remove	until this service is restarted.\
				them.\
			----- ----- ----- ----- ----- ----- ----- ----- ----- -----	vm-scaler
Periodically requests that vm-service provision	VM instances for machine and Remote Docker	![Moderate](images/circle-warning.png){width=20px}	vm-service-db-migrator \	
	more instances for running machine and remote Docker jobs.\	might not be provisioned causing you to run out		
			of capacity to run jobs with these executors.\	

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
	----- ----- ----- ----- ----- ----- ----- ----- ----- -----	vm-service	Inventory of available vm-service instances, and	Jobs that use machine or remote Docker will
![[Moderate]](images/circle-warning.png){width=20px}	vm-service-db-migrator \			provisioning of new instances. \
fail. \			----- ----- ----- ----- ----- ----- ----- ----- ----- -----	vm-service-db-migrator
Used to run database migrations for vm-service. \	Only runs at startup. \	![[Mild]](images/circle-success.png){width=20px}	None \	
----- ----- ----- ----- ----- ----- ----- ----- ----- -----	workflows-conductor	Coordinates and provides information about	No new workflows will start, currently running	![[Severe]](images/circle-failure.png){width=20px}

Container / Image	Role	What happens if it fails?	Failure severity	Startup dependencies
postgres \			workflows. \	workflows might end up in an inconsistent state,
	frontend \			
and some REST and GraphQL API requests will		workflows-conductor-migrator \		
	fail. \			----- ----- ----- ----- ----- ----- ----- ----- ----- -----
workflows-conductor-migrator	Runs postgresSQL migrations for the	Only runs on startup. \	! [Mild] (images/circle-success.png) {width=20px}	postgres \
		workflows-conductor. \		

|Metric |Description

|circle.nomad.server_agent.poll_failure |Returns 1 if the last poll of the Nomad agent failed, otherwise it returns 0.

|circle.nomad.server_agent.jobs.pending |Returns the total number of pending jobs across the cluster.

|circle.nomad.server_agent.jobs.running |Returns the total number of running jobs across the cluster.

|circle.nomad.server_agent.jobs.complete |Returns the total number of complete jobs across the cluster.

|circle.nomad.server_agent.jobs.dead |Returns the total number of dead jobs across the cluster.


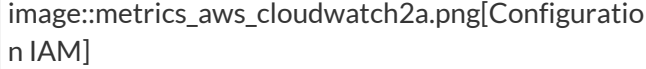
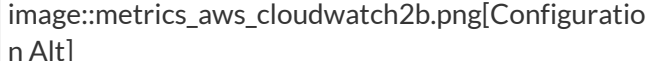
When the Nomad metrics container is running normally, no output will be written to standard output or standard error. Failures will elicit a message to standard error.

== Supported Platforms

We have two built-in platforms for metrics and monitoring: AWS CloudWatch and DataDog. The sections below detail enabling and configuring each in turn.

=== AWS CloudWatch

To enable AWS CloudWatch complete the following:

1. Navigate to the settings page within your Management Console. You can use the following URL, substituting your CircleCI URL: + https://<your-circleci-hostname>.com:8800/settings#cloudwatch_metrics `
 2. Check Enabled under AWS CloudWatch Metrics to begin configuration. + [AWS CloudWatch]
- ==== AWS CloudWatch Configuration
- There are two options for configuration:
- * Use the IAM Instance Profile of the services box and configure your custom region and namespace. + [Configuration IAM]
 - * Alternatively, you may use your AWS Access Key and Secret Key along with your custom region and namespace. + [Configuration Alt]

After saving you can **verify** that metrics are forwarding by going to your AWS CloudWatch console.

=== DataDog

```
sudo tee /etc/replicated.conf (cat <<'EOF'  
HTTP_PROXY=<proxy-ip:port>  
HTTPS_PROXY=<proxy-ip:port>
```

```
EOF
```

<pre>sudo tee -a /etc/circle-installation-customizations sudo service replicated-ui stop; sudo service replicated stop; sudo service replicated-operator stop; sudo service replicated-ui start; sudo service replicated-operator start; sudo service replicated start `</pre> <p>If you run in Amazon's EC2 service then you'll need to include 169.254.169.254 EC2 services as shown below:</p> <pre>` echo '{"HttpProxy": "http://<proxy-ip:port>"}'</pre>	<pre>sudo tee /etc/replicated.conf (cat <<'EOF' HTTP_PROXY=<proxy-ip:port> HTTPS_PROXY=<proxy-ip:port> NO_PROXY=169.254.169.254,<circleci-service-ip>, 127.0.0.1,localhost,ghe.example.com JVM_OPTS="- Dhttp.proxyHost=<ip> -Dhttp.proxyPort=<port> -Dhttps.proxyHost=<proxy-ip> -Dhttps.proxyPort=<port> -Dhttp.nonProxyHosts=169.254.169.254</pre>
<circleci-service-ip>	127.0.0.1
localhost	ghe.example.com"
	EOF

```
sudo tee -a /etc/circle-installation-customizations
sudo service replicated-ui stop; sudo service
replicated stop; sudo service replicated-operator
stop; sudo service replicated-ui start; sudo service
replicated-operator start; sudo service replicated
start `
```

NOTE: The above is not handled by by our enterprise-setup script and will need to be added to the user data for the Services Machine startup or done manually.

=== Corporate Proxies

WARNING: When our instructions ask if you use a proxy, you will also be prompted to input the address. It is **very important** that you input the proxy in the following format: `<protocol>://<ip>:<port>`. If you miss any part, then apt-get won't work correctly and the packages won't download.

=== Nomad Client Configuration

==== External Network Calls

CircleCI uses curl and awscli scripts to download initialization scripts, along with jars from Amazon S3. Both curl and awscli respect environment settings, but if you have whitelisted traffic from Amazon S3 you should not have any problems.

==== Internal Network Calls

* CircleCI JVM: **Any connections to other Nomad Clients or the Services machine should be excluded from HTTP proxy** Connections to GitHub Enterprise should be excluded from HTTP proxy

<circleci-service-ip>

127.0.0.1	localhost
-----------	-----------

ghe.example.com" EOF)

```
sudo tee -a /etc/environment
```

set -a ./etc/environment ` + You will also need to follow the Docker instructions to make sure your containers have outbound/proxy access.

=== Troubleshooting

If you cannot access the CircleCI Management Console, but the Services machine seems to be running, try to SSH tunnel into the machine by running the following, substituting your proxy address and the IP address of your Services machine:

```
shell ssh -L 8800:<address you want to proxy through>:8800
ubuntu@<ip_of_services_machine> ``
```

== Data Persistence Contact support@circleci.com to discuss externalizing services for data persistence.

= Authentication :page-layout: classic-docs :page-liquid: :icons: font :toc: macro :toc-title:

This document describes how to enable, configure, and test CircleCI to authenticate users with OpenLDAP or Active Directory credentials.

NOTE: LDAP is not supported with existing installations, only clean installations may use LDAP.

toc::[]

== Prerequisites

- * Install and configure your LDAP server and Active Directory.
- * GitHub Enterprise must be configured and is the source of organizations and projects to which users have access.
- * Install a new instance of CircleCI 2.0 with no existing users, following our [\[install\]](#) guide.
- * Contact [CircleCI support](#) and file a feature request for CircleCI Server.

NOTE: After completing this configuration, all users must log in to CircleCI with their LDAP credentials.

| Need | Path | More info

| General Config | [/etc/circle-installation-customizations](#) | See table below for values

| JVM Heap Sizes | [/etc/circleconfig/XXXX/customizations](#) Supports: frontend, test_results |
Adjust heap size for individual containers with [JVM_HEAP_SIZE](#)

| Custom CA Certs | [/usr/local/share/ca-certificates/](#) |

| Container Customizations | [/etc/circleconfig/XXX/customizations](#) | Used lots of places in replicated

| [/etc/hosts](#) | [/etc/hosts](#) | Respected by several containers including frontend, copied to container's /etc/hosts

| [/etc/environment](#) | [/etc/environment](#) | Respected by all containers

==== Properties of [/etc/circle-installation-customizations](#)

NOTE: Every property should be in the format `export ENV_VAR="value"`

[.table.table-striped] [cols=3*, options="header", stripes=even]

| Property | Impact | More info

| CIRCLE_URL | Override the scheme and host that CircleCI uses |

| JVM_HEAP_SIZE | Set JVM heap size for **all** containers reading this property | Use container specific settings when possible (see files above)

==== Other Properties and Env Vars

[.table.table-striped] [cols=3*, options="header", stripes=even]

| Property | Impact | More info

| HTTP_PROXY, NO_PROXY | Proxy for replicated and other services outside CircleCI containers to use |

== On Demand and Preallocated Instances Remote Docker and **machine** executor instances are spun up on demand. It is also possible to preallocate instances to remain up and running, ready for remote Docker and **machine** jobs to be run (see the last two fields in figure 9).

WARNING: If [Docker Layer Caching \(DLC\)](#) is to be used, VM Service instances must be on-demand so both remote Docker and **machine** preallocated instance fields must be set to **0**.

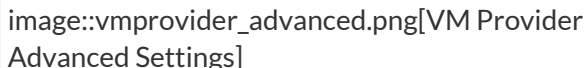
NOTE: When using preallocated instances be aware that a cron job is scheduled to cycle through these instances once per day to ensure they don't end up in an unworkable state.

== Job and Instance Management

Jobs run using the remote Docker Environment, or the **machine** executor are scheduled and dispatched by the Nomad server to your Nomad clients and passed on to remote Docker or **machine** from there. This means jobs run on remote Docker and the **machine** executor can be monitored in the usual way, using the Nomad CLI. See our [Introduction to Nomad Cluster Operation](#) for more about Nomad commands and terminology.

NOTE: A cron job is scheduled to cycle all default and preallocated instances at least once per day to ensure instances don't end up in a dead/bad state.

== Accessing Remote Docker and **machine** instances
By default, private IP addresses are used to communicate with VM service instances. If you need to grant wider access, for example, to allow developers SSH access, this can be set using the checkbox in the VM Provider Advanced Settings.

.Allowing Access to VM Service Instances


= Setting Up Certificates :page-layout: classic-docs
:page-liquid: :icons: font :toc: macro :toc-title:

This document provides a script for using a custom Root Certificate Authority and the process for using

```
sed -ne '/-BEGIN CERTIFICATE-/,/-END  
CERTIFICATE-/p' > /usr/local/share/ca-  
certificates/ghe.crt`
```

Then, navigate to the system console at port 8800 and change the protocol to upgraded. You can change the protocol to HTTPS (TLS/SSLEnabled) setting and restart the services. When trying Test GitHub Authentication you should get Success now rather than x509 related error.

== Setting up ELB Certificates

CircleCI requires the following steps to get ELB (Elastic Load Balancing) certificates working as your primary certs. The steps to accomplish this are below. You will need certificates for the ELB and CircleCI Server as described in the following sections.

NOTE: Opening the port for HTTP requests will allow CircleCI to return a HTTPS redirect.

1. Open the following ports on your ELB: +
[.table.table-striped] [cols=6*, options="header", stripes=even]

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	Cipher	SSL Certificate
HTTP	80	HTTP	80	N/A	N/A
SSL	443	SSL	443	Change	your-cert
SSL	3000	SSL	3000	Change	your-cert
HTTPS	8800	HTTPS	8800	Change	your-cert
SSL	8081	SSL	8081	Change	your-cert
SSL	8082	SSL	8082	Change	your-cert

2. Add the following security group on your ELB: + NOTE: The sources below are left open so that anybody can access the instance over these port ranges. If that is not what you want, then feel free to restrict them. Users will experience reduced functionality if your stakeholders are using IP addresses outside of the Source Range.

+ [.table.table-striped][cols=4*, options="header", stripes=even]

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0
HTTPS	TCP	443	0.0.0.0
Custom TCP Rule	TCP	8800	0.0.0.0
Custom TCP Rule	TCP	64535-65535	0.0.0.0

3. Next, in the management console for CircleCI, upload a valid certificate and key file to the **Privacy** Section. These don't need to be externally signed or even current certs as the actual cert management is done at the ELB. But, to use HTTPS requests, CircleCI requires a certificate and key in which the "Common Name (FQDN)" matches the hostname configured in the admin console.

4. It is now possible to set your Github Authorization Callback to **https** rather than **http**.

=== Using Self-Signed Certificates

Because the ELB does not require a *current* certificate, you may choose to generate a self-signed certificate with an arbitrary duration.

1. Generate the certificate and key using openssl command **openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 1 -out certificate.pem**

2. Provide the appropriate information to the prompts. + NOTE: The Common Name provided must match the host configured in CircleCI.

3. Save the certificate.pem and key.pem file locally.

== Setting up TLS/HTTPS on CircleCI Server

You may use various solutions to generate valid SSL certificate and key file. Two solutions are provided below.

=== Using Certbot

This section describes setting up TLS/HTTPS on your Server install using Certbot by manually adding a DNS record set to the Services machine. Certbot generally relies on verifying the DNS record via either port 80 or 443, however this is not supported on CircleCI Server installations as of 2.2.0 because of port conflicts.

. Stop the Service CircleCI Server Management Console (<http://<circleci-hostname>.com:8800>).

. SSH into the Services machine.

. Install Certbot and generate certificates using the following commands: + **shell sudo apt-get update sudo apt-get install software-properties-common sudo add-apt-repository ppa:certbot/certbot sudo apt-get update sudo apt-get install certbot certbot certonly --manual --preferred-challenges dns**

4. You will be instructed to add a DNS TXT record.

5. After the record is successfully generated, save **fullchain.pem** and **privkey.pem** locally.

If you are using Route 53 for your DNS records, adding a TXT record is straightforward. When you're creating a new record set, be sure to select type **TXT** and provide the appropriate value enclosed in quotes.

=== Adding the certificate to CircleCI Server

| Category | Safe Type

| Text | Plain

| Application | json

| Image | png

| Image | jpg

| Image | gif

| Image | bmp

| Video | webm

| Video | ogg

| Video | mp4

| Audio | webm

| Audio | aac

| Audio | mp4

| Audio | mpeg

| Audio | ogg

| Audio | wav

Also, by default, the following types will be rendered as plain text:

```
[.table.table-striped] [cols=2*,options="header",stripes=even]
```

| Category | Type

| Text | html

| Text | css

| Text | javascript

| Text | ecmascript

| Application | javascript

| Application | ecmascript

| Text | xml

== Allow Unsafe Content types If you would like to allow content types that are not included in the whitelist above, follow these steps:

1. Navigate to the CircleCI Management Console (e.g. <https://<your-circleci-hostname>:8800/settings>) and select Settings from the menu bar 2. Scroll down to find the Artifacts section 3. Select Serve Artifacts with Unsafe Content-Types + .Allow Unsafe Content Types image::UnsafeContentTypes.png[Build Artifacts] 4. Click Save at the bottom of the page and Restart Now in the pop-up to save your changes and restart the console.

WARNING: Any change to the settings within the Management Console will incur downtime as the console will need to be restarted.

= Enabling Usage Statistics :page-layout: classic-docs :page-liquid: :icons: font :toc: macro :toc-title:

This chapter is for System Administrators who want to automatically send some aggregate usage statistics to CircleCI. Usage statistics data enhances visibility into CircleCI installations and is used to better support you and ensure a smooth transition from CircleCI 1.0 to CircleCI 2.0.

toc::[]

To opt-in to this feature, navigate to your Management Console settings (e.g. <http://<circleci-hostname>.com:8800/settings>) and scroll down to Usage Statistics. Enable the radio button labeled Automatically send some usage statistics to CircleCI, as shown below.

image::usage-statistics-setting.png[Send Usage Statistics]

== Detailed Usage Statistics

The following sections provide information about the usage statistics CircleCI will gather when this setting is enabled.

=== Weekly Account Usage

[.table.table-striped] [cols=3*,options="header",stripes=even]

Name	Type	Purpose
------	------	---------

account_id	UUID	Uniquely identifies each vcs account
------------	------	--------------------------------------

usage_current_macos	minutes	For each account, track weekly builds performed in minutes.
---------------------	---------	---

usage_legacy_macos	minutes	
--------------------	---------	--

usage_current_linux	minutes	
---------------------	---------	--

usage_legacy_linux	minutes	
--------------------	---------	--

=== Weekly Job Activity

[.table.table-striped][cols=3*,options="header",stripes=even]

| Name | Type | Purpose

| utc_week | date | *Identifies which week the data below applies to*

| usage_oss_macos_legacy | minutes | *Track builds performed by week*

| usage_oss_macos_current | minutes |

| usage_oss_linux_legacy | minutes |

| usage_oss_linux_current | minutes |

| usage_private_macos_legacy | minutes |

| usage_private_macos_current | minutes |

| usage_private_linux_legacy | minutes |

| usage_private_linux_current | minutes |

| new_projects_oss_macos_legacy | sum | *Captures new Builds performed on 1.0. Observe if users are starting new projects on 1.0.*

| new_projects_oss_macos_current | sum |

| new_projects_oss_linux_legacy | sum |

| new_projects_oss_linux_current | sum |

| new_projects_private_macos_legacy | sum |

| new_projects_private_macos_current | sum |

| new_projects_private_linux_legacy | sum |

| new_projects_private_linux_current | sum |

| projects_oss_macos_legacy | sum | *Captures Builds performed on 1.0 and 2.0. Observe if users are moving towards 2.0 or staying with 1.0.*

| projects_oss_macos_current | sum |

| projects_oss_linux_legacy | sum |

| projects_oss_linux_current | sum |

| projects_private_macos_legacy | sum |

| projects_private_macos_current | sum |


```
| projects_private_linux_legacy | sum |
```

```
| projects_private_linux_current | sum |
```

== Accessing Usage Data If you would like programatic access to this data in order to better understand your users you may run this command from the Services VM.

```
shell docker exec usage-stats /src/builds/extract
```

=== Security and Privacy

Please reference exhibit C within your terms of contract and our [standard license agreement](#) for our complete security and privacy disclosures.

= Backup and Recovery :page-layout: classic-docs :page-liquid: :icons: font :toc: macro :toc-title:

This chapter describes failover or replacement of the services machine. Refer to the Backup section below for information about possible backup strategies and procedures for implementing a regular backup image or snapshot of the services machine.

toc::[]

== Disaster Recovery Specify a spare machine, in an alternate location, with the same specs for disaster recovery of the services machine. Having a hot spare regularly imaged with the backup snapshot in a failure scenario is best practice.

At the very least, provide systems administrators of the CircleCI installation with the hostname and location (even if co-located) of an equivalent server on which to install a replacement server with the latest snapshot of the services machine configuration. To complete recovery, use the Installation procedure, replacing the image from that procedure with your backup image.

== Backing up CircleCI Data

This document describes how to back up your CircleCI application so that you can recover from accidental or unexpected loss of CircleCI data attached to the Services machine:

NOTE: If you are running CircleCI in an HA configuration, you must use standard backup mechanisms for the external datastores. Contact support@circleci.com for more information document for more information.

== Backing up the Database

If you have **not** configured CircleCI for external services, the best practice for backing up your CircleCI data is to use VM snapshots of the virtual disk acting as the root volume for the Services machine. Backups may be performed without downtime as long the underlying virtual disk supports such an operation as is true with AWS EBS. There is a small risk, that varies by filesystem and distribution, that snapshots taken without a reboot may have some data corruption, but this is rare in practice.

NOTE: "Snapshots Disabled" refers to Replicated's built-in snapshot feature that is turned off by default.

== Backing up Object Storage

Build artifacts, output, and caches are generally stored in object storage services like AWS S3. These services are considered highly redundant and are unlikely to require separate backup. An exception is if your instance is setup to store large objects locally on the Services machine, either directly on-disk or on an NFS volume. In