



INSTALLATION GUIDE

A guide for installing and upgrading CircleCI Server on AWS.

docs@circleci.com

Version 2.19.9, 11/18/2020: FINAL

Installation Overview	1
Support Packages	1
CircleCI Server v2.19.x What's New	2
Requirements for Upgrading	2
Notes and Best Practices	2
Known Issues	2
What's New in v2.19.09	3
What's New in v2.19.08	3
What's New in v2.19.07	3
What's New in v2.19.06	3
What's New in v2.19.05	3
What's New in v2.19.04	3
What's New in v2.19.03	4
What's New in v2.19.02	4
What's New in v2.19.01	5
What's New in v2.19	5
Nomad Client Launch Configuration Update	6
IMPORTANT	6
Updated Nomad Client Launch Configuration Script (Last Update: July 10, 2020)	8
System Requirements	11
Services Machine	11
Nomad Clients	11
Choosing Nomad Client Quantity	12
Server Ports	12
Installation Prerequisites	18
Private Subnet Requirements	18
Planning	18
Installation on AWS with Terraform	21
Define Variables for Terraform	21
Provision Instances	23
Access Your Installation	24
Installation Setup	27
Validate Your Installation	31
Teardown	33
Upgrading a Server Installation	34
Nomad Launch Configuration	34
Org Rename Script	34
Upgrade Steps Overview	34

Installation Overview

The following sections provide planning information, system requirements and step-by-step instructions for installing CircleCI Server on Amazon Web Services (AWS) with Terraform.

Refer to the [What's New](#) page for full details of what's new and fixed in this release.

If you are looking to update an existing installation, see our guide to [Upgrading a Server Installation](#).

Support Packages

CircleCI 2.0 may be installed without a support package, on AWS, using the examples and instructions in this document. Alternatively, if you do decide to go ahead with a support package, there are a number of benefits, as detailed below:

Non-AWS Platform Support

With a Platinum CircleCI support package it is possible to install and configure CircleCI on Azure or any other platform used in your organization. Contact [CircleCI support](#) or your account representative to get started.

Externalization

With a Platinum support agreement, it is possible to improve performance and resilience by configuring the following services to run externally to the Services machine:

- PostgreSQL
- MongoDB
- Vault
- Rabbitmq
- Redis
- Nomad

Contact [CircleCI support](#) or your account representative to evaluate your installation against the current requirements for running external services.

CircleCI Server v2.19.x What's New

This document provides a summary of features and product notes for the release of CircleCI Server v2.19.9. For a full list of changes, including patch releases, refer to the [changelog](#). For a step by step guide to upgrading your CircleCI Server installation to v2.19.9, see our [upgrade guide](#).

Requirements for Upgrading



Before upgrading to 2.19.3, if you are using an IAM role scoped to a non-root path, you will need to unset the `OUTPUT_PROCESSOR_USE_NAIVE_ROLE_MAPPING` environment variable in your output processor customization script. See the [Customizations Guide](#) in our documentation for more information on using customization scripts.



For AWS installs, **before upgrading** to v2.19, follow [this guide](#) to update your nomad launch configuration.



If you are upgrading from pre v2.18.x, and have at any time changed your organization name, there is a [script](#) that **must** be run before starting the upgrade process. If you are already running v2.18.x, you will have run this already.

Notes and Best Practices

- We now require a minimum 32GB of RAM for the Services Machine.
- We made some changes to our Redis configuration in v2.18. If you have externalized Redis then you'll need to update your configuration. Please contact your Customer Success Manager if you are upgrading from pre v2.18 to v2.19.
- We have made changes to our Postgres version and require at least postgresSQL v9.5.16. If you have externalized postgresSQL then please update to at least that version in 2.17.x before upgrading to 2.19.9.

Known Issues

- On static (non-AWS) installations with `NONE` selected under storage driver settings, tests can be split only by file name or file size. If a user attempts to split tests by timing data, static instances will split them by file name instead.
- **If you are using an HTTP proxy for your installation, upgrade to v2.19.02 or above.** We had a known issue around job step logging and docker commands customization that made pre-2.19.2 releases incompatible with an HTTP proxy setup of CircleCI Server.
- The first user to access a CircleCI Server installation should be granted admin access. Currently, there is an issue preventing this from happening when LDAP authentication is in use. Follow [this guide](#) for a workaround.
- If any changes have been made to your networking configuration from the default, you may need to run the following steps to ensure you can use SSH to inspect your builds:
 - On all Nomad Client machine, create `/etc/circleci/public-ipv4`
 - This file should contain the public (if applicable) or private IP of the nomad client



For more information on SSH reruns in Server installations, see the [SSH Rerun Architecture in Server](#) guide.

- Classic Load Balancer is no longer available from v2.19.02 due to the ciphersuite changes listed in the notes below. CircleCI no longer accepts requests from [Classic Load Balancer](#), so you should move to [Network Load Balancer \(NLB\)](#) or [Application Load Balancer \(ALB\)](#).

What's New in v2.19.09

- Better handling of non-alphanumeric characters in authentication passwords for Docker executors. Fixed a bug that was preventing the use of `~` and `?` in the [Docker ID password field under the `auth key`](`{{ '/2.0/private-images/#docker-executor' | docs_url }}`).
- Improved the way commit statuses are handled to reduce instances of them being published out of order in certain situations.
- General security updates.

What's New in v2.19.08

- Fixed a bug that was causing parallelism to fail for static installations with `NONE` selected under storage driver settings.

What's New in v2.19.07

- Fixed a bug that was preventing authentication to China region ECR images in the docker executor.
- Removed the recursive `chown` from startup of the `fileservd` service. In instances of heavy usage this was causing the startup process to take a long time, or in some cases, startup was blocked.

What's New in v2.19.06

- Fixed a bug that was causing workflow statuses to be displayed on GitHub in the wrong order.
- Introduced performance improvements to `workflows-conductor` that dramatically reduce CPU usage and latency.
- Fixed a bug that was preventing use of an S3 storage region other than `us-east-1` via an IAM user.
- Fixed formatting type of SMTP passwords to ensure they are masked during setup.

What's New in v2.19.05

- Fixed a bug that could lead to the VM database ending up in an incorrect state in the event of a Services Machine crash or manual termination of VM Service instances.

What's New in v2.19.04

- Fixed a bug that was leading to support bundle creation timing out before the Services machine logs had been created, leaving only Replicated logs.
- S3 connection pool metrics under `circle.s3.connection_pool.*` have been added to the test results service, making it easier to debug issues with this service.

- Added in missing environment variables for the workflows service. The absence of these environment variables was causing excessive stack tracing whenever workflows were run. This in turn lead to excessive log rotation.
- Fixed a bug that was causing failure to update GitHub statuses. Some customers experienced this bug when a project had a follower with a broken auth token.

What's New in v2.19.03

- Removed the use of the depeccated GitHub.com API endpoint `GET applications/%s/tokens/%s`.
- Distributed tracing is now enabled by default for Server installations. Traces are used in CircleCI support bundles to improve our ability to troubleshoot Server issues. Options for the tracing sampling rate are displayed in the Replicated Management Console, but should only be changed from the default if requested by CircleCI Support.
- Fixed an issue that was preventing `restore_cache` from working with the storage driver set to "none" - i.e not S3.
- Fixed an issue that was preventing the `output_processor` service from using AWS AssumeRole when the role was located in a subfolder. This issue affected customers with security policies forcing the use of a subfolder in this case, and the symptoms included the inability to store artifacts or use timings-based test splitting.
- JVM heap size can now be changed using the `JVM_HEAP_SIZE` environment variable for the following services: `vm-service`, `domain-service`, `permissions-service` and `federations-service`.

What's New in v2.19.02

- In the LDAP login flow we now use an anonymous form to `POST` LDAP auth state, rather than sending it as a `GET` parameter. Previously, when a user authenticated using LDAP, their username and password were sent in plaintext as part of a query parameter in a `GET` request. As requests are over HTTPS, this left usernames and passwords in request logs, etc. This issue is now fixed.
- Optimizely and Zendesk are now removed from Server release images.
- Fixed an issue in which setting `CIRCLE_ADMIN_SERVER_HTTP_THREADS` or `CIRCLE_PUBLIC_FACING_SERVER_HTTP_THREADS` too high would prevent the frontend container from starting.
- Due to changes in the GitHub API we have removed the use of `?client_id=x&client_secret=y` for GitHub, and GHE versions 2.17 and later.
- Fixed an issue that was causing intermittent failures to spin up VMs with DLC in use.
- Fixed an issue that was preventing the customization of proxy settings for Docker containers. See the [Nomad Client Proxy](#) and [Service Configuration Overrides](#) guides for more infomation.
- Fixed a bug that was preventing job steps for non-failing builds being logged when proxy settings were used for the job container.
- Removed legacy TLS versions 1.0 and 1.1, in addition, enabled 1.2 and 1.3 TLS, and specified the following ciphersuites
 - ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384

- Fixed a `statsd` configuration issue that meant some services were not emitting Telegraf metrics.

What's New in v2.19.01

- Fixed a bug that was preventing some customers from upgrading due to a schema change in one of our library dependencies.
- Fixed a bug that was preventing some customers from inspecting builds via SSH due to a logic change in our build agent.

What's New in v2.19

- You can now customize resource classes for your installation to provide developers with [CPU/RAM options](#) for the Jobs they configure. For more information [see our guide to customizing resource classes in Server v2.19](#).
- CircleCI Server installations on AWS can now be [configured to work on GovCloud](#).
- The image used to run the RabbitMQ server has been updated to fix vulnerabilities.

Nomad Client Launch Configuration Update

IMPORTANT



This update must be run before upgrading to Server v2.19

- **THIS WILL REQUIRE DOWNTIME.** Please schedule a maintenance window before proceeding.
- This change is only compatible with CircleCI Server versions greater than or equal to 2.19.0
- If you are using the Static version of CircleCI Server, you will need to recreate your builders manually. Refer to the guide linked [here](#) for the updated scripts and further instruction.

Sign in to AWS

1. Login to the AWS
2. Select the region that your CircleCI Server (Server) resides in
3. Select the `Services` tab and select `EC2`

Find the Launch Configuration and Service Box IP address

1. Select `Auto Scaling Groups` located near the bottom of the left navigation pane
2. Select the auto scaling group configuration associated with your Server installation. Typically resembles `*-ghe_nomad_clients_asg`
3. Make note of the name of `Launch Configuration` under the `Details` tab. Typically resembles `terraform-20200114212820082700000001`
4. In the left navigation pane, select `Launch Configurations`
5. Locate the launch configuration that matches the one found in step 3.
6. In the bottom pane, select the ``View User Data`` link
7. Scroll down or search for the `Creating config.hcl` section.
8. Write down or copy the service box private IP address located in the client portion of the config located next to `servers = ["IP_ADDRESS:4647"]` You will need this in the next step.

```
...
...
client {
  enabled = true
  # Expecting to have DNS record for nomad server(s)
  servers = [ "YOUR_SERVICES_BOX_IP_ADDRESS:4647" ] <==== YOU WILL NEED THIS IP ADDRESS
  node_class = "linux-64bit"
  options = {"driver.raw_exec.enable" = "1"}
}
...
```


Create a New Launch Configuration

1. Right-click the launch configuration and select `Copy Launch Configuration`
2. At the top of the screen, select `3. Configure details`
3. Update the name to something meaningful and easy to identify IE `nomad-client-lc-20200117`
4. Select the `Advanced Details` dropdown
5. Below you will find the updated Nomad client configuration. Replace the contents of the `User data` pane with the script below.
6. **IMPORTANT:** Locate the section of code you found in step 8 above and replace the IP address with that of your services box.
7. Select the `Skip to review` button at the bottom right of the screen
8. Select the `Create Launch Configuration Button`
9. Verify that the SSH key-pair name is correct and select `Create launch configuration` button

Update the Auto Scaling group

1. Return the `Auto Scaling Groups` page
2. Locate the auto scaling group we identified in the `Find the Launch Configuration and Service Box IP address` section above
3. Right-click the auto scaling group and select `Edit` from the menu
4. Select the launch configuration created in the previous section from the `Launch Configuration` dropdown menu.
5. Press the `Save` button

Rolling the Nomad Client Instances

There are many strategies you can use to update the builders. Some ideas include but are not limited to:

- Terminate all of the existing nomad client instances and allow the auto scaler to recreate them.
- Edit the auto scaling configuration and set the `Desired Capacity` to 0. Once the existing instances have terminated, set the `Desired Capacity` to the original number.

Please use whatever works best with your existing methods of auto scaling group management.

Verify the New Nomad Clients are Communicating with Nomad Server

1. SSH into the CircleCI Services Box
2. Enter the following command: `sudo docker exec -it nomad nomad node-status`. The output should resemble the following:

```
ubuntu@govcloud-service-box:~$ sudo docker exec -it nomad nomad node-status
```

ID	DC	Name	Class	Drain	Eligibility	Status
0cf07b07	default	i-070fdad5f0edef4c8	linux-64bit	false	eligible	ready
ec2ccc9d	us-east-1	i-0895ee505ec7e692c	linux-64bit	false	eligible	down

3. Verify that there are new builders containing the `default` DC name and they are in a `ready` state.

Updated Nomad Client Launch Configuration Script (Last Update: July 10, 2020)

```
#!/bin/bash

set -exu

export http_proxy=""
export https_proxy=""
export no_proxy=""
export aws_instance_metadata_url="http://169.254.169.254"
export PUBLIC_IP="$(curl $aws_instance_metadata_url/latest/meta-data/public-ipv4)"
export PRIVATE_IP="$(curl $aws_instance_metadata_url/latest/meta-data/local-ipv4)"
export DEBIAN_FRONTEND=noninteractive
UNAME="$(uname -r)"

echo "-----"
echo "      Performing System Updates"
echo "-----"
apt-get update && apt-get -y upgrade

echo "-----"
echo "      Installing NTP"
echo "-----"
apt-get install -y ntp

# Use AWS NTP config for EC2 instances and default for non-AWS
if [ -f /sys/hypervisor/uuid ] && [ `head -c 3 /sys/hypervisor/uuid` == ec2 ]; then
cat <<EOT > /etc/ntp.conf
driftfile /var/lib/ntp/ntp.drift
disable monitor

restrict default ignore
restrict 127.0.0.1 mask 255.0.0.0
restrict 169.254.169.123 nomodify notrap

server 169.254.169.123 prefer iburst
EOT
else
echo "USING DEFAULT NTP CONFIGURATION"
fi

service ntp restart
```

```

echo "-----"
echo "      Installing Docker"
echo "-----"
apt-get install -y apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs)
stable"
apt-get install -y "linux-image-$UNAME"
apt-get update
apt-get -y install docker-ce=5:18.09.9~3-0~ubuntu-xenial

# force docker to use userns-remap to mitigate CVE 2019-5736
apt-get -y install jq
mkdir -p /etc/docker
[ -f /etc/docker/daemon.json ] || echo '{}' > /etc/docker/daemon.json
tmp=$(mktemp)
cp /etc/docker/daemon.json /etc/docker/daemon.json.orig
jq '.["userns-remap"]="default"' /etc/docker/daemon.json > "$tmp" && mv "$tmp"
/etc/docker/daemon.json

sudo echo 'export http_proxy="${http_proxy}"' >> /etc/default/docker
sudo echo 'export https_proxy="${https_proxy}"' >> /etc/default/docker
sudo echo 'export no_proxy="${no_proxy}"' >> /etc/default/docker
sudo service docker restart
sleep 5

echo "-----"
echo "      Populating /etc/circleci/public-ipv4"
echo "-----"
if ! (echo $PUBLIC_IP | grep -qP "^[\\d.]+$")
then
    echo "Setting the IPv4 address below in /etc/circleci/public-ipv4."
    echo "This address will be used in builds with \"Rebuild with SSH\"."
    mkdir -p /etc/circleci
    echo $PRIVATE_IP | tee /etc/circleci/public-ipv4
fi

echo "-----"
echo "      Installing nomad"
echo "-----"
apt-get install -y zip
curl -o nomad.zip https://releases.hashicorp.com/nomad/0.9.3/nomad_0.9.3_linux_amd64.zip
unzip nomad.zip
mv nomad /usr/bin

echo "-----"
echo "      Creating config.hcl"
echo "-----"
export INSTANCE_ID="$(curl $aws_instance_metadata_url/latest/meta-data/instance-id)"
mkdir -p /etc/nomad
cat <<EOT > /etc/nomad/config.hcl
log_level = "DEBUG"
name = "$INSTANCE_ID"

```

```

data_dir = "/opt/nomad"
datacenter = "default"
advertise {
  http = "$PRIVATE_IP"
  rpc = "$PRIVATE_IP"
  serf = "$PRIVATE_IP"
}
client {
  enabled = true
  # Expecting to have DNS record for nomad server(s)
  servers = ["REPLACE_ME_WITH_SERVICE_BOX_IP:4647"]
  node_class = "linux-64bit"
  options = {"driver.raw_exec.enable" = "1"}
}

```

EOT

```

echo "-----"
echo "      Creating nomad.conf"
echo "-----"
cat <<EOT > /etc/systemd/system/nomad.service
[Unit]
Description="nomad"
[Service]
Restart=always
RestartSec=30
TimeoutStartSec=1m
ExecStart=/usr/bin/nomad agent -config /etc/nomad/config.hcl
[Install]
WantedBy=multi-user.target

```

EOT

```

echo "-----"
echo "      Creating ci-privileged network"
echo "-----"
docker network create --driver=bridge --opt com.docker.network.bridge.name=ci-privileged ci-privileged

echo "-----"
echo "      Starting Nomad service"
echo "-----"
service nomad restart

```

System Requirements

This section defines the system and port access requirements for installing CircleCI v2.19.9.

Services Machine

The Services machine hosts the core of our Server product, including the user-facing website, API engine, datastores, and Nomad job scheduler. It is best practice to use an isolated machine.



Currently, CircleCI Server only supports x86_64 architecture.

The following table defines the Services machine CPU, RAM, and disk space requirements:

Number of daily active CircleCI users	CPU	RAM	Disk space	NIC speed
<50	8 cores	32GB	100GB	1Gbps
50-250	12 cores	64GB	200GB	1Gbps
251-1000	16 cores	128GB	500GB	10Gbps
1001-5000	20 cores	256GB	1TB	10Gbps
5000+	24 cores	512GB	2TB	10Gbps

Nomad Clients

Nomad client machines run CircleCI jobs that are scheduled by the Nomad Server. The Minimum CPU, RAM, and disk space requirements per client are as follows:

- CPU: 4 cores
- RAM: 32GB
- Disk space: 100GB
- NIC speed: 1Gbps

For an AWS install of CircleCI Server, the recommended instance type for Nomad clients is `m5.2xlarge` (8 vCPUs @ 2.4GHz, 32GB RAM).



Currently, Nomad Clients only support x86_64 architecture.

You can choose a larger instance type to fit more jobs per Client. To help in this choice, consider that when Nomad decides if a job will fit on a Client, the Job is allocated 1024MHz per CPU, and capacity is `number of cores * clock speed`. Using this method, the maximum capacity of a `m5.2xlarge` would be 19200MHz, which would mean 9.6 jobs could run on that client (if there were no limiting factors). In practice, Nomad will reserve some CPU for itself, and because of the CPU:RAM ratio, the available RAM is the limiting factor governing how many jobs can run.

Recommendations for Heavy Usage

EBS will throttle IOPS after certain IO operations, therefore heavy-load usage with the default Server configuration (m5.2xlarge with 200-GiB gp2 root volume) can see performance-related issues, such as slow jobs, slow spin-up, and job timeouts in worst cases.

To give an idea of what is meant by heavy usage:

- If `BurstBalance` metrics for storage attached to Nomad clients are decreasing sharply, this is a sign that the workload is heavy enough to be affected. See the [AWS EBS metrics document](#) for further details.
- We have had a report from a customer that continuous disk I/O at 500 MB/s used up the burst credits for our default storage configuration within 37.5 minutes.

Suggested workarounds for this issue are as follows:

- Use EBS volumes larger than 1 TiB. Large EBS volumes never consume IO credits.
- Use the `*5d` instance type and use an attached instance store for Docker-related files (i.e. `/var/lib/docker`).

Choosing Nomad Client Quantity

The following table defines the number of Nomad clients to make available as a best practice. Scale up and down according to demand on your system:

Number of daily active CircleCI users	Number of Nomad client machines
<50	1-5
50-250	5-10
250-1000	10-15
5000+	15+

Server Ports

Below all ports required by a CircleCI 2.0 installation are listed for each machine type.

Services Machine

Port number	Protocol	Direction	Source / destination	Use	Notes
80	TCP	Inbound	End users	HTTP web app traffic	
443	TCP	Inbound	End users	HTTPS web app traffic	
7171	TCP	Inbound	End users	Artifacts access	
8081	TCP	Inbound	End users	Artifacts access	

Port number	Protocol	Direction	Source / destination	Use	Notes
22	TCP	Inbound	Administrators	SSH	
8800	TCP	Inbound	Administrators	Admin console	
8125	UDP	Inbound	Nomad Clients	Metrics	
8125	UDP	Inbound	Nomad Servers	Metrics	Only if using externalized Nomad Servers
8125	UDP	Inbound	All Database Servers	Metrics	Only if using externalised databases
4647	TCP	Bi-directional	Nomad Clients	Internal communication	
8585	TCP	Bi-directional	Nomad Clients	Internal communication	
7171	TCP	Bi-directional	Nomad Clients	Internal communication	
3001	TCP	Bi-directional	Nomad Clients	Internal communication	
80	TCP	Bi-directional	GitHub Enterprise / GitHub.com (whichever applies)	Webhooks / API access	
443	TCP	Bi-directional	GitHub Enterprise / GitHub.com (whichever applies)	Webhooks / API access	
80	TCP	Outbound	AWS API endpoints	API access	Only if running on AWS
443	TCP	Outbound	AWS API endpoints	API access	Only if running on AWS
5432	TCP	Outbound	PostgreSQL Servers	PostgreSQL database connection	Only if using externalised databases. Port is user-defined, assuming the default PostgreSQL port.

Port number	Protocol	Direction	Source / destination	Use	Notes
27017	TCP	Outbound	MongoDB Servers	MongoDB database connection	Only if using externalized databases. Port is user-defined, assuming the default MongoDB port.
5672	TCP	Outbound	RabbitMQ Servers	RabbitMQ connection	Only if using externalized RabbitMQ
6379	TCP	Outbound	Redis Servers	Redis connection	Only if using externalized Redis
4647	TCP	Outbound	Nomad Servers	Nomad Server connection	Only if using externalized Nomad Servers
443	TCP	Outbound	CloudWatch Endpoints	Metrics	Only if using AWS CloudWatch

Nomad Clients

Port number	Protocol	Direction	Source / destination	Use	Notes
64535-65535	TCP	Inbound	End users	SSH into builds feature	
80	TCP	Inbound	Administrators	CircleCI Admin API access	
443	TCP	Inbound	Administrators	CircleCI Admin API access	
22	TCP	Inbound	Administrators	SSH	
22	TCP	Outbound	GitHub Enterprise / GitHub.com (whichever applies)	Download Code From GitHub.	
4647	TCP	Bi-directional	Services Machine	Internal communication	
8585	TCP	Bi-directional	Services Machine	Internal communication	
7171	TCP	Bi-directional	Services Machine	Internal communication	
3001	TCP	Bi-directional	Services Machine	Internal communication	
443	TCP	Outbound	Cloud Storage Provider	Artifacts storage	Only if using external artifacts storage
53	UDP	Outbound	Internal DNS Server	DNS resolution	This is to make sure that your jobs can resolve all DNS names that are needed for their correct operation.

GitHub Enterprise / GitHub.com

Port number	Protocol	Direction	Source / destination	Use	Notes
22	TCP	Inbound	Services Machine	Git access	
22	TCP	Inbound	Nomad Clients	Git access	
80	TCP	Inbound	Nomad Clients	API access	
443	TCP	Inbound	Nomad Clients	API access	
80	TCP	Bi-directional	Services Machine	Webhooks / API access	

PostgreSQL Servers

Port number	Protocol	Direction	Source / destination	Use	Notes
5432	TCP	Bi-directional	PostgreSQL Servers	PostgreSQL replication	Only if using externalized databases. Port is user-defined, assuming the default PostgreSQL port.

MongoDB Servers

Port number	Protocol	Direction	Source / destination	Use	Notes
27017	TCP	Bi-directional	MongoDB Servers	MongoDB replication	Only if using externalized databases. Port is user-defined, assuming the default MongoDB port.

RabbitMQ Servers

Port number	Protocol	Direction	Source / destination	Use	Notes
5672	TCP	Inbound	Services Machine	RabbitMQ connection	Only if using externalized RabbitMQ
5672	TCP	Bi-directional	RabbitMQ Servers	RabbitMQ mirroring	Only if using externalized RabbitMQ

Redis Servers

Port number	Protocol	Direction	Source / destination	Use	Notes
6379	TCP	Inbound	Services Machine	Redis connection	Only if using externalized Redis
6379	TCP	Bi-directional	Redis Servers	Redis replication	Only if using externalized Redis, and using Redis replication (optional)

Nomad Servers

Port number	Protocol	Direction	Source / destination	Use	Notes
4646	TCP	Inbound	Services Machine	Nomad Server connection	Only if using externalized Nomad Servers
4647	TCP	Inbound	Services Machine	Nomad Server connection	Only if using externalized Nomad Servers
4648	TCP	Bi-directional	Nomad Servers	Nomad Servers internal communication	Only if using externalized Nomad Servers

Installation Prerequisites

CircleCI uses Terraform to automate parts of the infrastructure for your CircleCI Server install, so you will need to install this first:

- Visit [Download Terraform](#) and choose the correct package for your architecture.

Ensure you have the following information available before beginning the installation procedure:

- A CircleCI License file (`.r1i`). Contact [CircleCI support](#) for a license and request a cluster-enabled license to run jobs on dedicated instances for best performance.
- Your AWS Access Key ID and Secret Access Key.
- Name of your [AWS EC2 key pair](#).
- [AWS Region](#), for example `us-west-2`.
- AWS Virtual Private Cloud (VPC) ID and AWS Subnet ID. If your account is configured to use a default VPC, your default VPC ID is listed under Account Attributes, which you will find from the AWS management console on the EC2 dashboard page.
- Set your VPC (`enableDnsSupport`) setting to `true` to ensure that queries to the Amazon provided DNS server at the 169.254.169.253 IP address, or the reserved IP address at the base of the VPC IPv4 network range plus two will succeed. See the [Using DNS with Your VPC](#) Amazon Web Services documentation for additional details.

Private Subnet Requirements

The following additional settings are required to support using private subnets on AWS with CircleCI:

- The private subnet for builder boxes must be configured with a [NAT gateway](#) or an [internet gateway](#) configured for the outbound traffic to the internet via attached route tables.



The subnet should be large enough to **never** exhaust the addresses.

- The [VPC Endpoint for S3](#) should be enabled. Enabling the VPC endpoint for S3 should significantly improve S3 operations for CircleCI and other nodes within your subnet.
- Adequately power the NAT instance for heavy network operations. Depending on the specifics of your deployment, it is possible for NAT instances to become constrained by highly parallel builds using Docker and external network resources. A NAT that is inadequate could cause slowness in network and cache operations.
- If you are integrating with [github.com](#), ensure that your network access control list (ACL) allows ports 80 and 443 for GitHub webhooks. When integrating with GitHub, either set up CircleCI in a public subnet, or set up a public load balancer to forward github.com traffic.
- See the [Services Machine](#) section of our overview for more information on the specific ports that need to be accessible to instances in your CircleCI installation.

Planning

Have available the following information and policies before starting the installation:

- If you use network proxies, contact your Account team before beginning your install.
- Plan to provision at least two AWS instances, one for Services and one for your first set of Nomad Clients. Best practice is to use an `m4.2xlarge` instance with 8 vCPUs and 32GB RAM for both the Services and Nomad Clients instances.
- AWS instances must have outbound access to pull Docker containers and to verify your license. If you don't want to give open outbound access, see our [list of ports](#) that will need access.
- In order to provision required AWS entities with Terraform you will require an IAM User with the following permissions (See the [AWS guidance](#) on creating IAM users):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:*"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::circleci-*",
        "arn:aws:s3:::circleci-*/*",
        "arn:aws:s3:::*"
      ]
    },
    {
      "Action": [
        "autoscaling:*",
        "sqs:*",
        "iam:*",
        "ec2:StartInstances",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "ec2:Describe*",
        "ec2:CreateTags",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateSecurityGroup",
        "ec2>DeleteSecurityGroup",
        "ec2:DescribeInstanceAttribute",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstances",
        "ec2:DescribeNetworkAcls",
```

```

        "ec2:DescribeSecurityGroups",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:ModifyInstanceAttribute",
        "ec2:ModifyNetworkInterfaceAttribute",
        "cloudwatch:*",
        "autoscaling:DescribeAutoScalingGroups",
        "iam:GetUser"
    ],
    "Resource": [
        "*"
    ],
    "Effect": "Allow"
}
]
}

```

Installation on AWS with Terraform

Following is a step by step guide to installing CircleCI Server v2.19.9 with Terraform.

Define Variables for Terraform

1. Clone the [Setup](#) repository. If you already have it cloned, make sure it is up-to-date and you are on the `master` branch by running:

```
git checkout master && git pull
```

2. Go to the top directory of the `enterprise-setup` repo on your local machine.
3. Run `terraform init` to initialize your working directory.
4. Run `make init` to initialize a `terraform.tfvars` file (your previous `terraform.tfvars` if any, will be backed up in the same directory).
5. Open `terraform.tfvars` in an editor and fill in appropriate AWS values for section 1.
6. If you plan to use 1.0 builders, specify a `circle_secret_passphrase` in section 2, replacing `...` with alpha numeric characters, if not, leave it as is. 1.0 builders are disabled by default in section 3.
7. Specify the instance type to use for your Nomad clients. By default, the value specified in the `terraform.tfvars` file for Nomad clients is `m5.2xlarge` (8 vCPUs, 32GB RAM). To increase the number of concurrent CircleCI jobs that each Nomad client can run, modify section 2 of the `terraform.tfvars` file to specify a larger `nomad_client_instance_type`. Refer to the AWS [Amazon EC2 Instance Types](#) guide for details, and read our guidance in the [System Requirements](#) document.



The `builder_instance_type` is only used for CircleCI 1.0 and is disabled by default in section 3.

8. In section 3 you can:
 - a. choose to use 1.0 Builders if your project requires it (by changing the count to 1)
 - b. enter proxy details, and enter a prefix if there will be multiple installations within your AWS region – the Services and Nomad client instances will be displayed with this prefix in the AWS console.

Example tfvars

```
#####
# 1. Required Cloud Configuration
#####

aws_access_key = "..."
aws_secret_key = "..."
aws_region = "eu-central-1"
aws_vpc_id = "..."
aws_subnet_id = "..."
aws_ssh_key_name = "..."

#####
# 2. Required CircleCI Configuration
#####

circle_secret_passphrase = "..."
services_instance_type = "m4.2xlarge"
builder_instance_type = "r3.4xlarge"
nomad_client_instance_type = "m4.2xlarge"

#####
# 3. Optional Cloud Configuration
#####

# Set this to `1` or higher to enable CircleCI 1.0 builders
desired_builders_count = "0"

# Provide proxy address if your network configuration requires it
http_proxy = ""
https_proxy = ""
no_proxy = ""

# Use this var if you have multiple installation within one AWS region
prefix = "..."
```

Above is an example of the `terraform.tfvars` file you will be editing. The table below shows some of the default settings, and some optional variables that can be used to further customize your cluster. A full list of variables and defaults can be found in the `variables.tf` file in the root of the `enterprise-setup` directory.



If you require your installation to work on AWS GovCloud, you can enable this by setting `enable_govcloud` to `true`.

Optional vars:

Var	Description	Default
<code>services_instance_type</code>	Instance type for the centralized services box. We recommend a m4 instance	m4.2xlarge
<code>builder_instance_type</code>	Instance type for the 1.0 builder machines. We recommend a r3 instance	r3.2xlarge
<code>max_builders_count</code>	Max number of 1.0 builders	2
<code>nomad_client_instance_type</code>	Instance type for the nomad clients (2.0 builders). We recommend a XYZ instance	m4.xlarge
<code>max_clients_count</code>	Max number of nomad clients	2
<code>prefix</code>	Prefix for resource names	circleci
<code>enable_nomad</code>	Provisions a nomad cluster for CircleCi Server v2.x	1
<code>enable_route</code>	Enable creating a Route53 route for the Services box	0
<code>enable_govcloud</code>	Allows deployment into AWS GovCloud	false
<code>services_user_data_enabled</code>	Set to 0 to disable automated installation on Services Box	1
<code>force_destroy_s3_bucket</code>	Add/Remove ability to forcefully destroy S3 bucket when your installation is shut down	false
<code>services_disable_api_termination</code>	Protect the services instance from API termination. Set to false if you would like to terminate the Services box automatically when your installation is shut down	true

Provision Instances

1. Save your changes to the `tfvars` file and run the following:

```
terraform plan
```

2. To provision your instances, run the following:

```
terraform apply
```

You will be asked to confirm if you wish to go ahead by typing `yes`.

3. An IP address will be provided at the end of the Terraform output. Visit this IP to carry on the install process.

Access Your Installation

1. Your browser may prompt you with a SSL/TLS info box. This is just to inform you that on the next screen your browser might tell you the connection to the admin console is unsafe, but you can be confident it is secure. Click Continue to Setup and proceed to your installation IP.

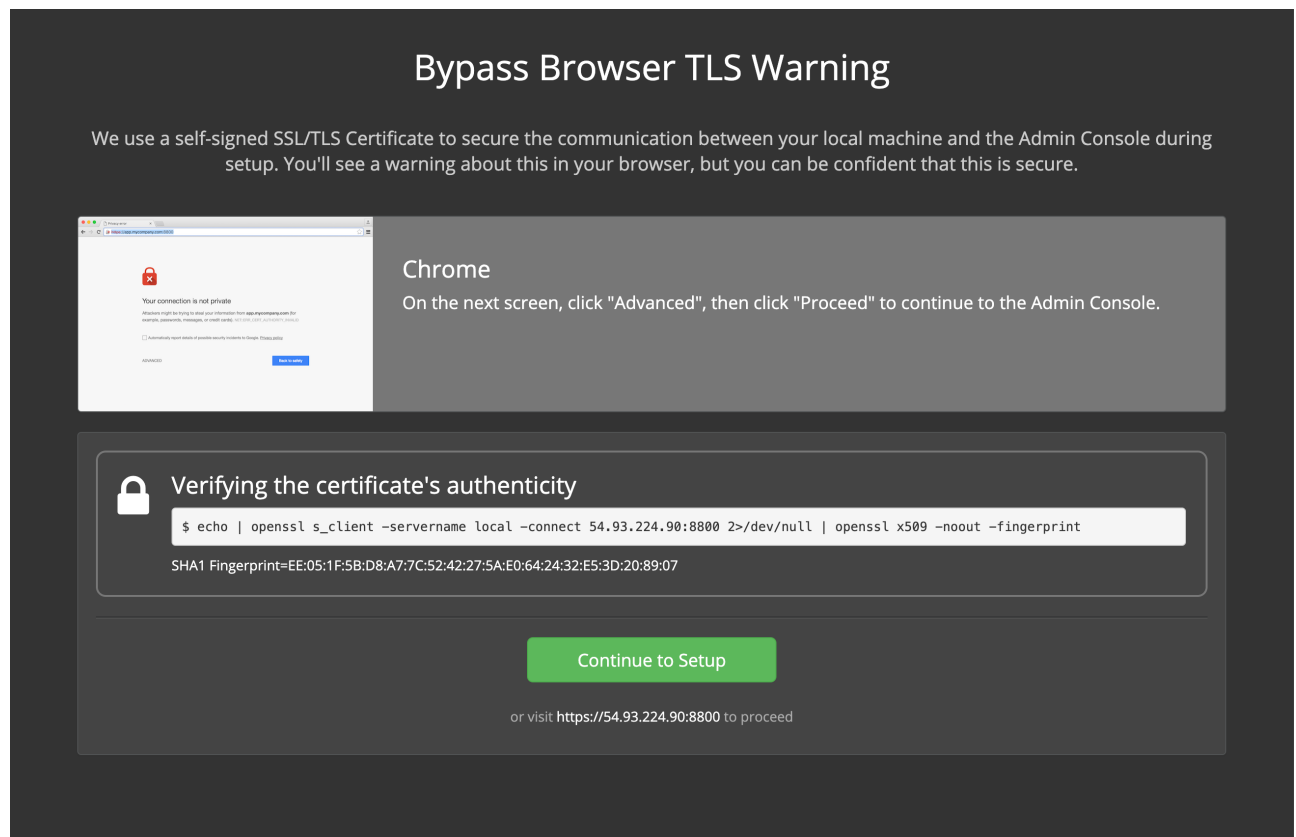


Figure 1. SSL Security

2. Enter your hostname. This can be your domain name or public IP of the Services Machine instance. At this time you can also upload your SSL public key and certificate if you have them. To proceed without providing these click Use Self-Signed Cert – choosing this option prompts security warnings each time you visit the Management Console.

Figure 2. Hostname

3. Upload your license.
4. Decide how to secure the Management Console. You have three options:
 - a. Anonymous admin access to the console, anyone on port 8800 can access (not recommended)
 - b. Set a password that can be used to securely access the Management Console (recommended)
 - c. Use your existing directory-based authentication system (for example, LDAP)

Secure the Admin Console

Keeping this admin console secure is important.
You can create a shared password that will be required to access the settings, or you can connect it to your existing directory based authentication system.

☐ Anonymous ☒ Password ☐ LDAP

Password

Confirm Password

Continue

Figure 3. Admin Password

5. Your CircleCI installation will be put through a set of preflight checks, once they have completed, scroll down and click Continue.

Preflight Checks

- ✓ **Successful HTTP request**
Can access api.replicated.com
- ✓ **OS linux is supported**
The operating system must be linux
- ✓ **Kernel version requirement met**
Kernel version must be at least 3.10
- ✓ **Successful TLS connection**
Can connect to TLS 172.31.23.155 address
- ✓ **Total space requirement met for directory /tmp**
Directory must have at least 1G total space
- ✓ **Total space requirement met for directory /var/lib/replicated**
Directory must have at least 250M total space
- ✓ **Docker server version requirement met**
Docker server version must be exactly 17.12.1
- ✓ **CPU cores requirement met**
Server must have at least 2 CPU cores
- ✓ **Memory requirement met**
Server must have at least 8G total memory
- ✓ **Total space requirement met for directory /**
Directory must have at least 60G total space
- ✓ **Total space requirement met for directory /var/lib/docker**
Directory must have at least 1G total space
- ✓ **Successful Docker registry ping**
Can access registry registry.replicated.com
- ✓ **Successful Docker registry ping**
Can access registry index.docker.io

Node: b210b1a2dd95...

- ✓ **OS linux is supported**
The operating system must be linux

Figure 4. Preflight Checks

Installation Setup

You should now be on the Management Console settings page (your-circleci-hostname.com:8800).



You can make changes to the settings on this page at any time but changes here will require **downtime** while the service is restarted. Some settings are covered in more detail in our Operations Guide.

1. **Hostname** – The Hostname field should be pre-populated from earlier in the install process, but if you skipped that step, enter your domain or public IP of the Services machine instance. You can check this has been entered correctly by clicking Test Hostname Resolution.
2. **Services** – The Services section is only used when externalizing services. Externalization is available with a Platinum service contract. Contact support@circleci.com if you would like to find out more.

Figure 5. External Services

3. **Execution Engines** – only select 1.0 Builders if you require them for a legacy project – most users will leave this unchecked.
4. **Builders Configuration** – select Cluster in the 2.0 section. The Single box option will run jobs on the Services machine, rather than a dedicated instance, so is only suitable for trialling the system, or for some small teams.

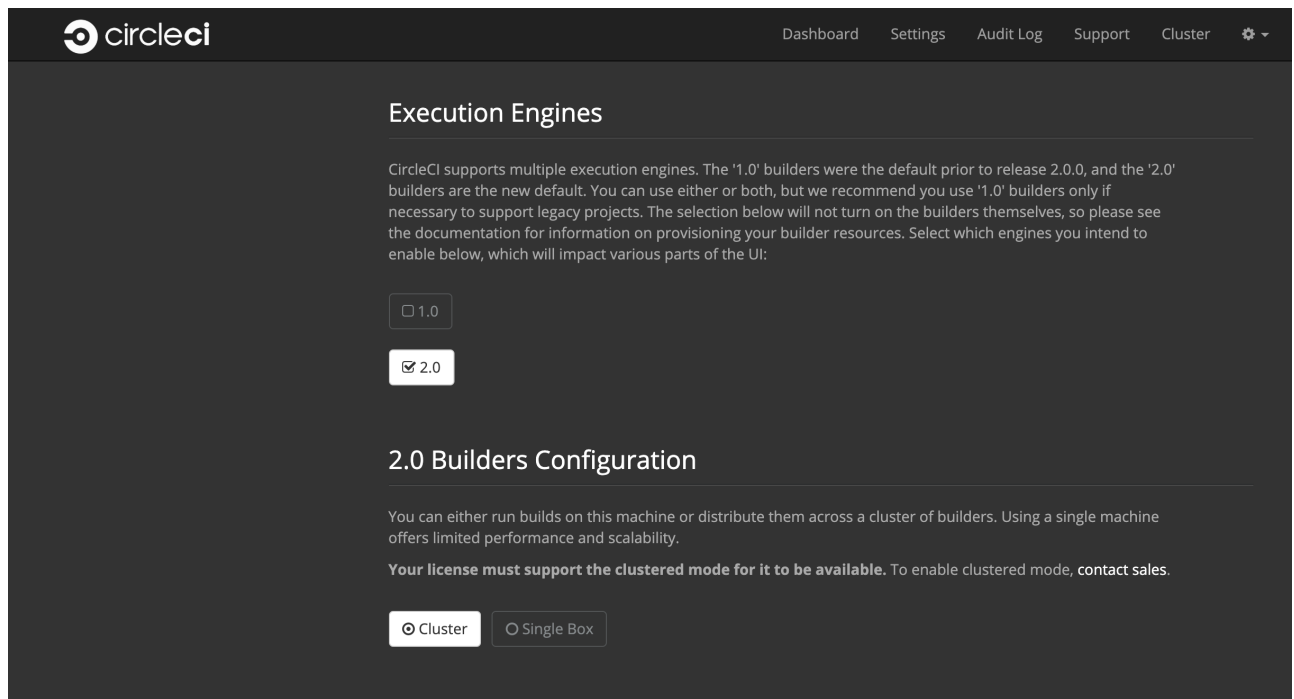


Figure 6. 1.0 and 2.0 Builders

5. **GitHub Integration** – register CircleCI as a new OAuth application in GitHub.com or GitHub Enterprise by following the instructions provided on the page.



If you get an "Unknown error authenticating via GitHub. Try again, or contact us." message, try using `http:` instead of `https:` for the Homepage URL and callback URL.

- a. Copy the Client ID and Secret from GitHub and paste it into the relevant fields, then click Test Authentication.
- b. If you are using GitHub.com, move on to step 6. If using Github Enterprise, you will also need to follow some supplementary steps and supply an API Token so we can verify your organization. To provide this, complete the following from your GitHub Enterprise dashboard:
 - i. Navigate to Personal Settings (top right) > Developer Settings > Personal Access Tokens.
 - ii. Click “generate new token”. Name the token appropriately to prevent accidental deletion. Do not tick any of the checkboxes, we only require the default public read-level access so no extra permissions are required. We recommend this token should be shared across your organization rather than being owned by a single user.
 - iii. Copy the new token and paste it into the GitHub Enterprise Default API Token field.

GitHub Integration

CircleCI integrates with github.com or GitHub Enterprise. You will need to create an "OAuth Application" for CircleCI by doing the following:

1. Go to <https://github.com/settings/applications/new> for github.com or <https://<your GitHub hostname>/settings/applications/new> for GitHub Enterprise
2. Enter an application name like "CircleCI"
3. Enter "http(s)://<your hostname>" as the homepage URL
4. Enter "http(s)://<your hostname>/auth/github" as the authorization callback URL.

NOTE: The hostname and http(s) protocol on the callback URL must match the hostname and SSL settings on this page. **NOTE:** CircleCI does not support changing the URL or Backend Github instance after it has been set. Doing so can potentially take down your instance of CircleCI. If this is a hard requirement then please contact us. Not following the instructions could lead to catastrophic system failure.

☐ Public GitHub ☒ GitHub Enterprise

GitHub Enterprise Domain (Required)

The hostname associated with GitHub Enterprise, e.g. ghe.example.com

☒ HTTPS (TLS/SSL Enabled) ☐ HTTP (unencrypted/insecure) ☐ HTTPS (with self signed)

GitHub Application Client ID (Required)

GitHub Application Client Secret (Required)

GitHub Enterprise Default API Token (Required)

A GitHub token to use for requests when no OAuth token is available.

Figure 7. Enter Github Enterprise Token

6. **LDAP** – if you wish to use LDAP authentication for your installation, enter the required details in the LDAP section. For a detailed runthrough of LDAP settings, read our [LDAP authentication guide](#)
7. **Privacy** – We recommend using an SSL certificate and key for your install. You can submit these in the Privacy section if this step was missed during the installation.

Privacy

☒ SSL only (Recommended)

Forces TLS/SSL for your installation. A valid x509 SSL certificate and private key files are required to use this option. The certificate and key must be in PEM format. The key must be *unencrypted*.

Certificate File (Required)

Private Key File (Required)

Figure 8. Privacy Settings

8. **Storage** – We recommend using S3 for storage and all required fields for Storage are pre-populated. The IAM user, as referred to in the [planning](#) section of this document, is used here.

Storage

CircleCI supports multiple cloud providers. We recommend using the native object storage of your provider to store build artifacts and files, which requires your cloud credentials for authenticating with the APIs. You can choose to use our experimental local object storage, but we do not recommend this for production systems.

☒ AWS S3 ☐ None

CircleCI will use an S3 bucket for storing build-related artifacts. Supply the AWS keys here.

To create an IAM user/role with the proper permissions, you can follow the [Getting Started - AWS document](#). (The provided Terraform or CloudFormation resources can automatically create an IAM role and instance profile.)

AWS Region (Required)

eu-central-1

S3 Bucket (Required)

rosie-bucket-44b34e2c

The bucket will be created if necessary.

☒ IAM Instance Profile ☐ IAM User (Key+Secret)

AWS Authentication

Figure 9. Storage Options

9. **Enhanced AWS Integration** – Complete this section if you are using 1.0 builders.
10. **Email** Complete the Email section if you wish to configure your own email server for sending build update emails. Leave this section is you wish to use our default email server.



Due to an issue with our third party tooling, Replicated, the Test SMTP Authentication button is not currently working

11. **VM Provider** – Configure VM service if you plan to use [Remote Docker](#) or `machine` executor (Linux/Windows) features. We recommend using an IAM instance profile for authentication, as described in the [planning](#) section of this document. With this section completed, instances will automatically be provisioned to execute jobs in Remote Docker or use the `machine` executor. To use the Windows `machine` executor you will need to [build an image](#). For more information on VM Service and creating custom AMIs for remote Docker and `machine` executor jobs, read our [VM service guide](#).

You can preallocate instances to always be up and running, reducing the time taken for Remote Docker and `machine` executor jobs to start. If preallocation is set, a cron job will cycle through your preallocated instances once per day to prevent them getting into a bad/dead state.



If Docker Layer Caching (DLC) is to be used, VM preallocation should be set to `0`, forcing containers to be spun up on-demand for both `machine` and Remote Docker. It is worth noting here that if these fields are **not** set to `0` but all preallocated instances are in use, DLC will work correctly, as if preallocation was set to `0`.

12. **AWS Cloudwatch or Datadog Metrics** can be configured for your installation. Set either of these up in the relevant sections. For more information read our [Monitoring guidance](#):

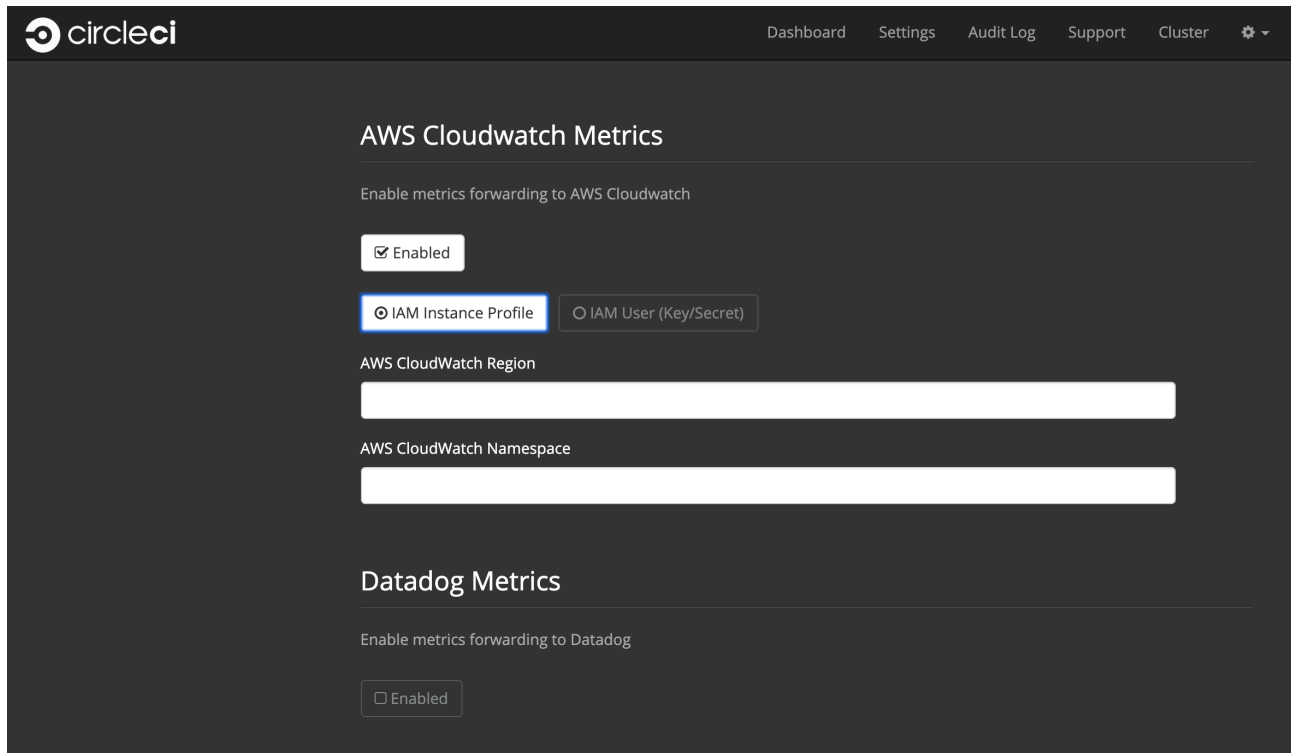
The screenshot shows the CircleCI Settings page for Metrics configuration. At the top, there's a navigation bar with links: Dashboard, Settings, Audit Log, Support, Cluster, and a gear icon. The main heading is "AWS Cloudwatch Metrics". Below it, there's a section "Enable metrics forwarding to AWS Cloudwatch" with a toggle switch set to "Enabled". There are two radio button options: "IAM Instance Profile" (which is selected and highlighted with a blue border) and "IAM User (Key/Secret)". Below these are two text input fields labeled "AWS CloudWatch Region" and "AWS CloudWatch Namespace". The second section is "Datadog Metrics", with a toggle switch set to "Disabled".

Figure 10. Metrics

13. **Custom Metrics** are an alternative to Cloudwatch and Datadog metrics, you can also customize the metrics you receive through Telegraf. For more on this read our [Custom Metics](#) guide.
14. **Distributed Tracing** is used in our support bundles, and settings should remain set to default unless a change is requested by CircleCI Support.
15. **Artifacts** persist data after a job is completed, and may be used for longer-term storage of your build process outputs. By default, CircleCI Server only allows approved types to be served. This is to protect users from uploading, and potentially executing malicious content. The **Artifacts** setting allows you to override this protection. For more information on safe/unsafe types read our [Build Artifacts guidance](#).
16. After agreeing to the License Agreement and saving your settings, select Restart Now from the popup. You will then be redirected to start CircleCI and view the Management Console Dashboard. It will take a few minutes to download all of the necessary Docker containers.



If the Management Console reports `Failure reported from operator: no such image` click Start again and it should continue.

Validate Your Installation

1. When the application is started, select Open to launch CircleCI in your browser, and sign up/log in to your CircleCI installation and start running 2.0 builds! You will become the Administrator at this point as you are the first person to sign in. Have a look at our [Getting Started](#) guide to start adding projects.

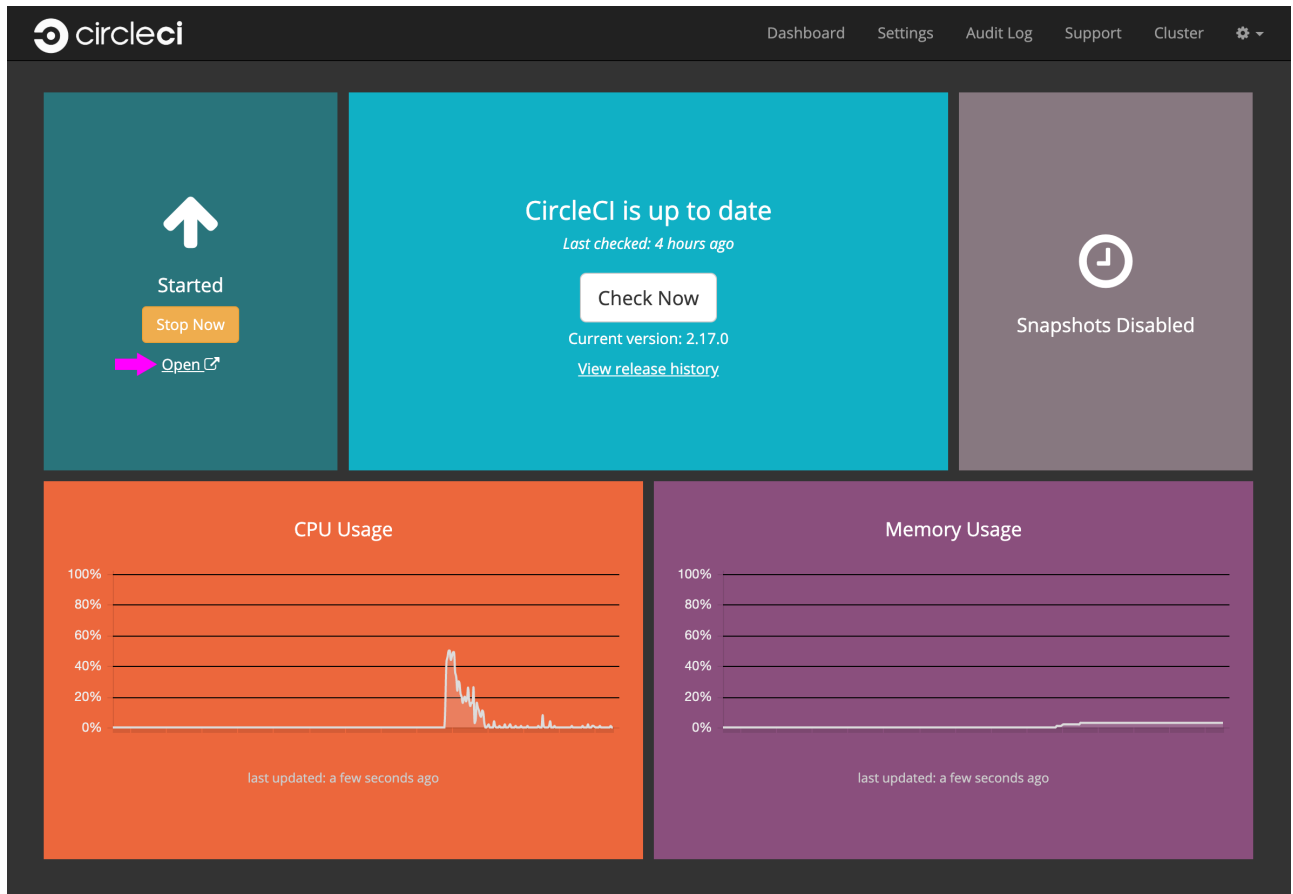


Figure 11. Start CircleCI from your Dashboard

2. After build containers have started and images have been downloaded, the first build should begin immediately. If there are no updates after around **15 minutes**, and you have clicked the Refresh button, contact [CircleCI support](#) for assistance.
3. Next, use [our realitycheck repo](#) to check basic CircleCI functionality.
4. If you're unable to run your first builds successfully please start with our [Troubleshooting](#) guide for general troubleshooting topics, and our [Introduction to Nomad Cluster Operation](#) for information about how to check the status of Builders in your installation.

Teardown

If you wish to delete your installation of CircleCI Server, please let us know first in case there are any specific, supplementary steps required for your installation. Below is our basic step by step guide to tearing down an installation of CircleCI Server that was made with Terraform:

1. First you need to manually disable the termination protection on the Services machine from the AWS Management Console (If you set `services_disable_api_termination = "false"` in your `terraform.tfvars` file, skip this step). To do this:
 - a. Navigate to the EC2 Dashboard and locate the Services machine instance
 - b. Click to select it
2. Click Actions > Instance Settings > Change Termination Protection
3. Navigate to the S3 dashboard, locate the S3 bucket associated with your CircleCI cluster and delete the bucket and its contents (If you set `force_destroy_s3_bucket = "true"` in your `terraform.tfvars` file, skip this step).
4. From a terminal, navigate to your clone of our `enterprise-setup` repo and run `terraform destroy` to destroy all EC2 instances, IAM roles, ASGs and Launch configurations created by `terraform apply`.

Upgrading a Server Installation

This document describes the process for upgrading your CircleCI Server installation to v2.19.9.

Nomad Launch Configuration



Before upgrading to v2.19, follow [this guide](#) to update your nomad launch configuration.

Org Rename Script



If upgrading directly from v2.17 (or below) and you have **ever had issues with renaming an organization within CircleCI** or you suspect that an **organization rename might have happened at any point** follow the steps below. If you are running Server v2.18.x already you can skip this section.

1. SSH into your Services machine
2. REPL into workflows-conductor by running the following: `sudo docker exec -it workflows-conductor lein repl :connect 6005`
3. Go to this link for the [org rename script](#). Copy/paste this script into the REPL session. It will run migration and output current progress.
4. If any `ERROR` messages are present in the output please report back to your CSM or reach out to support.

Upgrade Steps Overview

Follow these steps to upgrade your CircleCI Server version.

1. Snapshot for Rollback



This step is optional but recommended

To take a snapshot of your installation:

1. Go to the Management Console (e.g. `your-circleci-hostname.com:8800`) and click Stop Now to stop the CircleCI service.

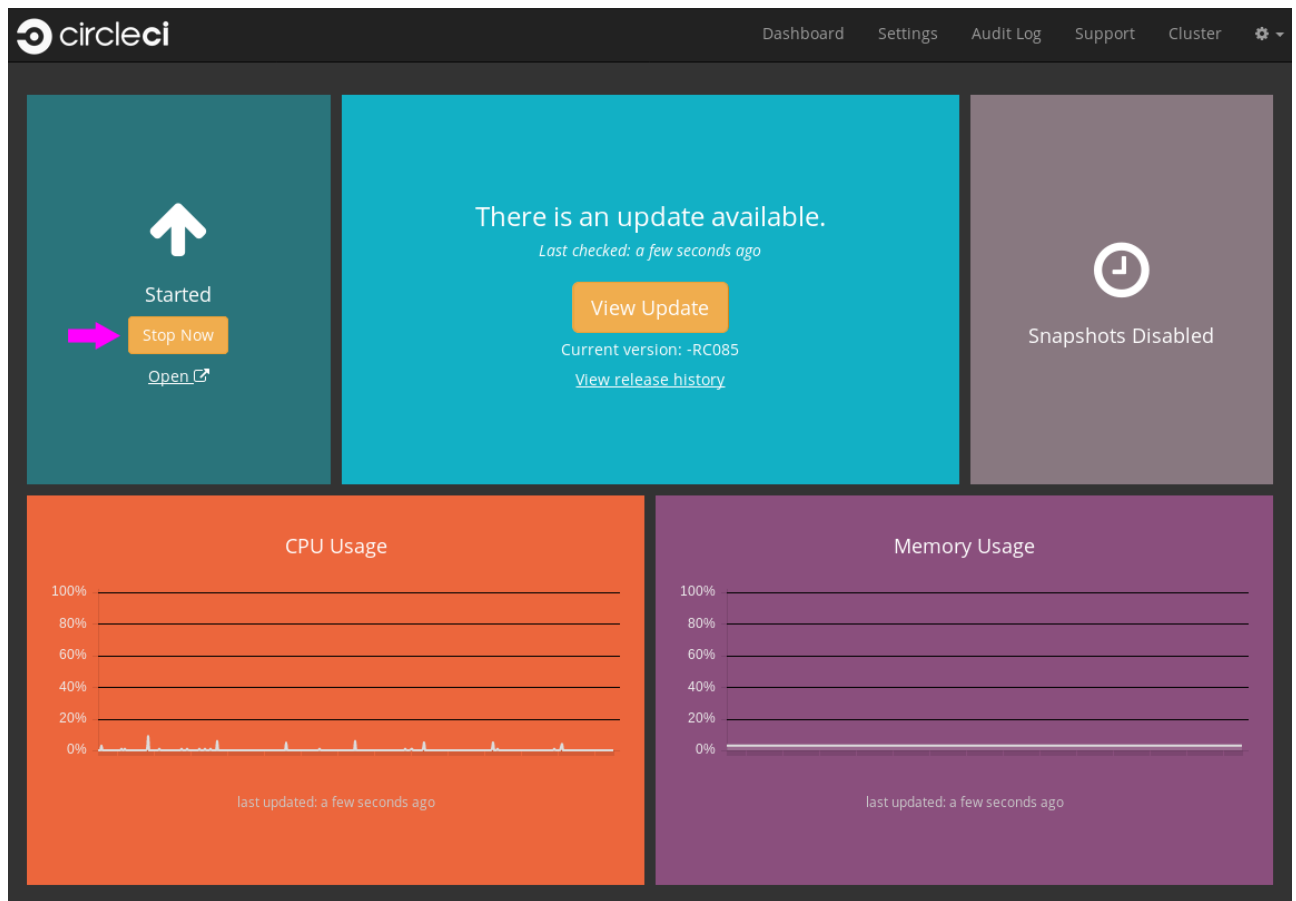


Figure 12. Stop CircleCI

2. Ensure no jobs are running on the nomad clients – you can check this by running `nomad status`
3. Navigate to the AWS EC2 management console and select your Services machine instance
4. Select Actions > Image > Create Image – Select the No Reboot option if you want to avoid downtime at this point. This image creation step creates an AMI that can be readily launched as a new EC2 instance to restore your installation.

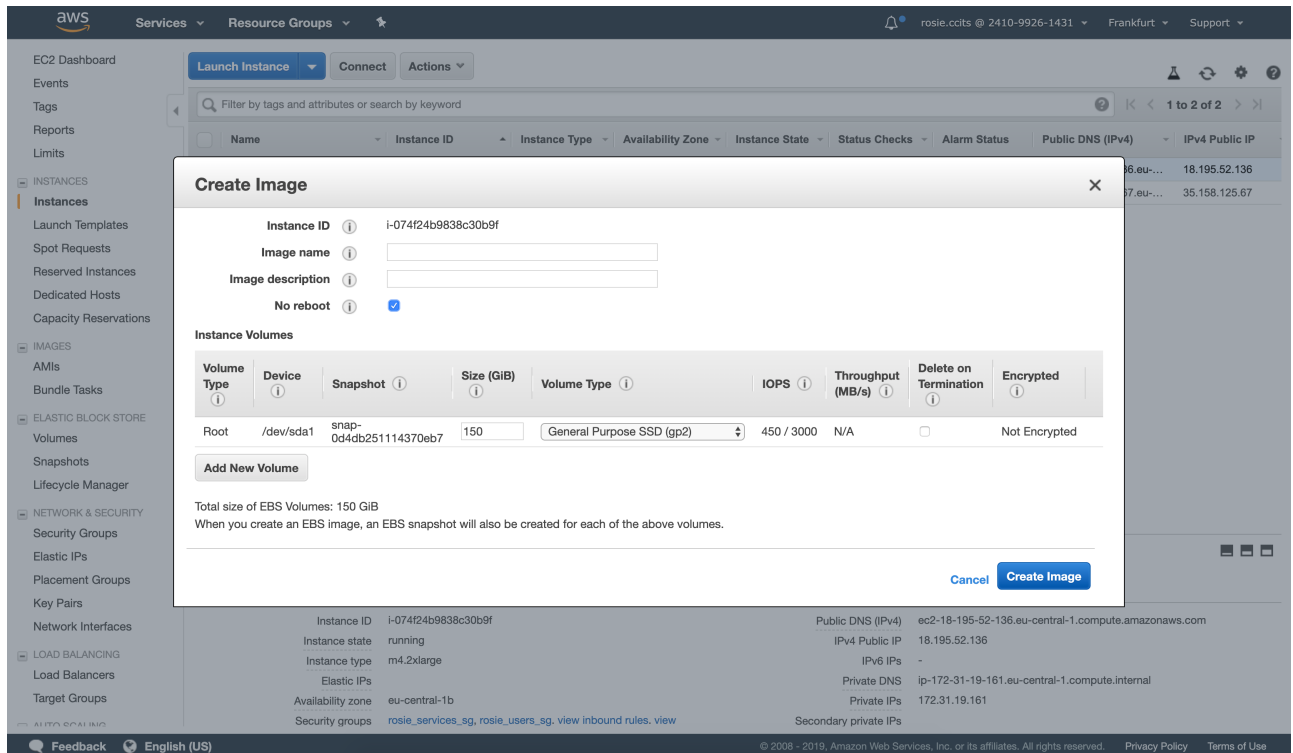


Figure 13. Snapshot Image Creation



It is also possible to automate this process with the AWS API. Subsequent AMIs/snapshots are only as large as the difference (changed blocks) since the last snapshot, such that storage costs are not necessarily larger for more frequent snapshots, see Amazon's EBS snapshot billing document for details. Once you have the snapshot you are free to make changes on the Services machine.

If you do need to rollback at any point, see our [guide to restoring from a backup](#).

2. Updating Replicated

a. Prerequisites

- Your installation is Ubuntu 14.04 or 16.04 based.
- Your installation is **not** airgapped and you can access the internet from it.
- We will be updating to Replicated v2.38.6, but first we need to check you are running at least v2.10.3 on your Services machine. To check this, SSH into the Services machine and run the following:

```
replicated --version
```

If you are running a version of Replicated pre v2.10.3 please reach out to support@circleci.com. If you are already on v2.38.6 or above you can skip the next step and move to [upgrade the CircleCI application](#)

b. Preparations



Remember to take a snapshot (described above) before starting the Replicated update process

1. Stop the CircleCI application by clicking the Stop Now button on the Dashboard. Application shutdown takes a few minutes. Wait for the status to become “Stopped” before continuing.

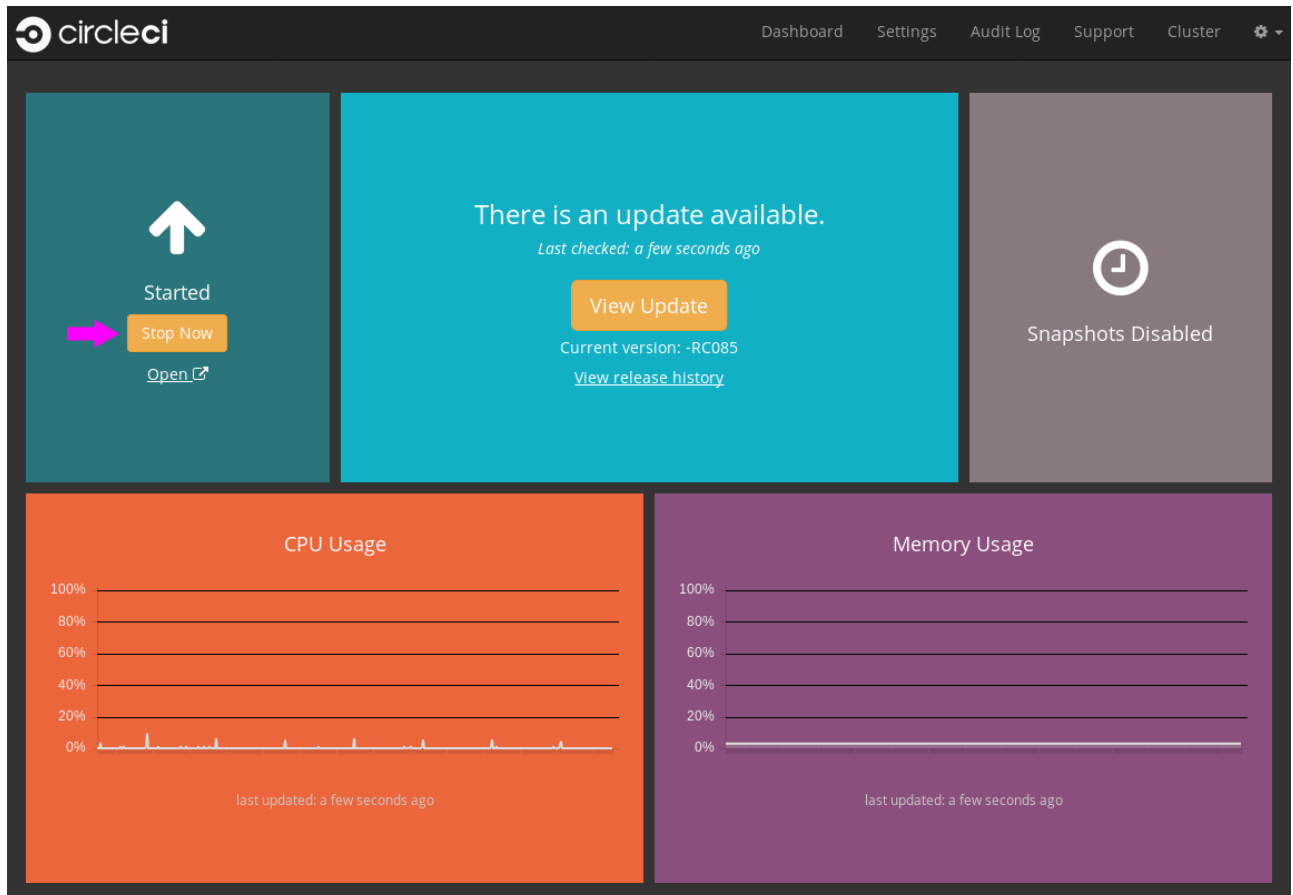


Figure 14. Stop the CircleCI Application

Alternatively you can SSH into the Services machine and stop the CircleCI application from the command line:

```
replicatedctl app stop
```

You can check the status using the following:

```
replicatedctl app status inspect
```

Example Output:

```
[
  {
    "AppID": "edd9471be0bc4ea04dfca94718ddf621",
    "Sequence": 2439,
    "State": "stopped",
    "DesiredState": "stopped",
    "Error": "",
    "IsCancellable": false,
    "IsTransitioning": false,
    "LastModifiedAt": "2018-10-23T22:00:21.314987894Z"
  }
]
```

- For the replicated update to succeed, it is necessary to update docker to the recommended version, 17.12.1. Check which version you are running with `docker version` and if you need to update, follow these steps:

```
sudo apt-get install docker-ce=17.12.1~ce-0~ubuntu
```

- Pin the Docker version using the following command:

```
sudo apt-mark hold docker-ce
```

c. Perform Update

- Perform the Replicated update by executing the update script as follows:

```
curl -sSL "https://get.replicated.com/docker?replicated_tag=2.38.6" | sudo bash
```

Double-check your replicated and docker versions:

```
replicatedctl version    # 2.38.6
docker -v                # 17.12.1
```

- Restart the app with

```
replicatedctl app start
```


The application will take a few minutes to spin up. You can check the progress in the administration dashboard or by executing:

```
replicatedctl app status inspect
```

Example output:

```
[
  {
    "AppID": "edd9471be0bc4ea04dfca94718ddf621",
    "Sequence": 2439,
    "State": "started",
    "DesiredState": "started",
    "Error": "",
    "IsCancellable": true,
    "IsTransitioning": true,
    "LastModifiedAt": "2018-10-23T22:04:05.00374451Z"
  }
]
```

3. Upgrade CircleCI Server

1. Once you are running the latest version of Replicated, click the View Update button in the Management Console dashboard.

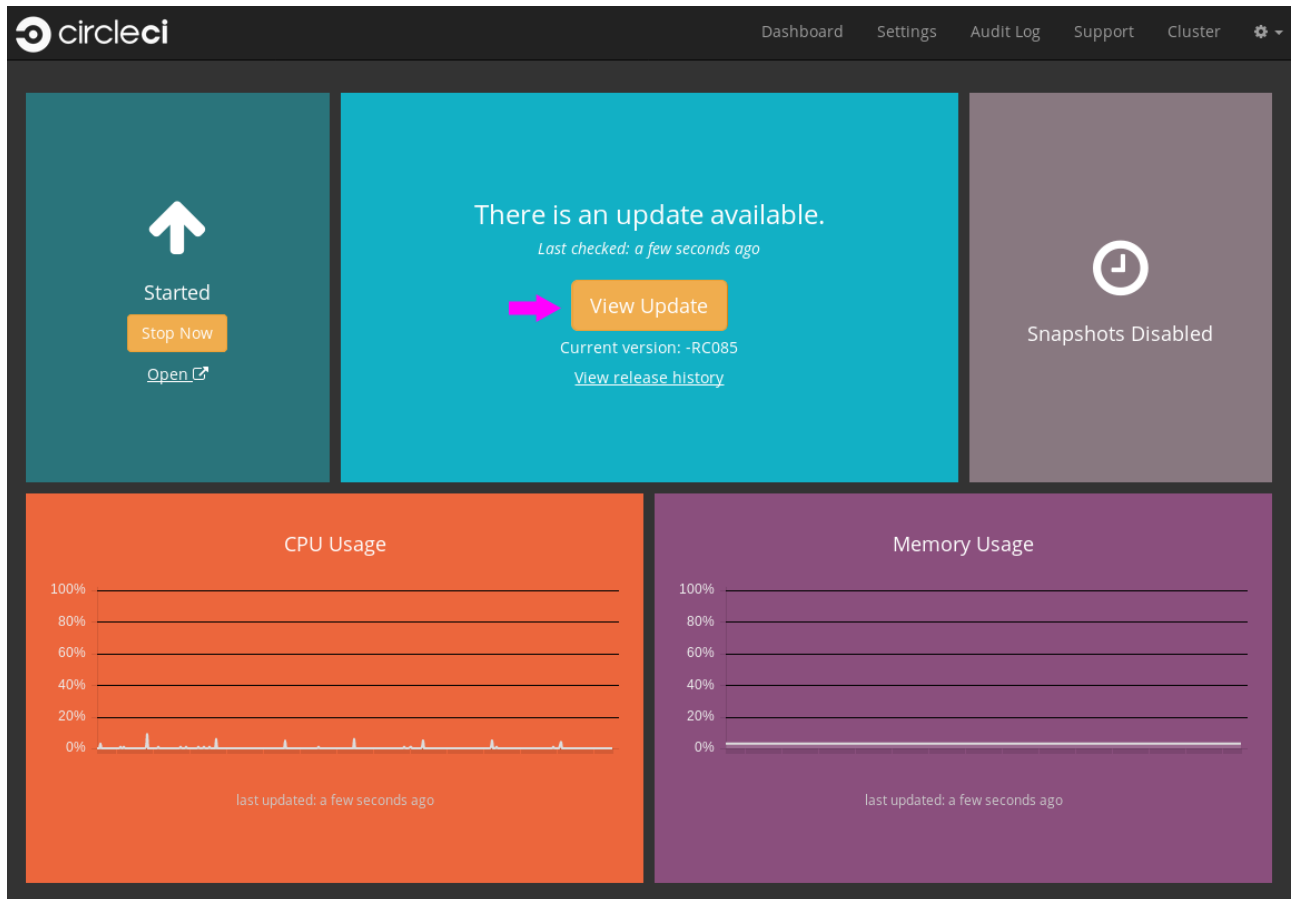


Figure 15. View Available Updates

2. Click Install next to the version you wish to install.



Please refresh your screen intermittently during the install process to avoid unnecessary waiting.

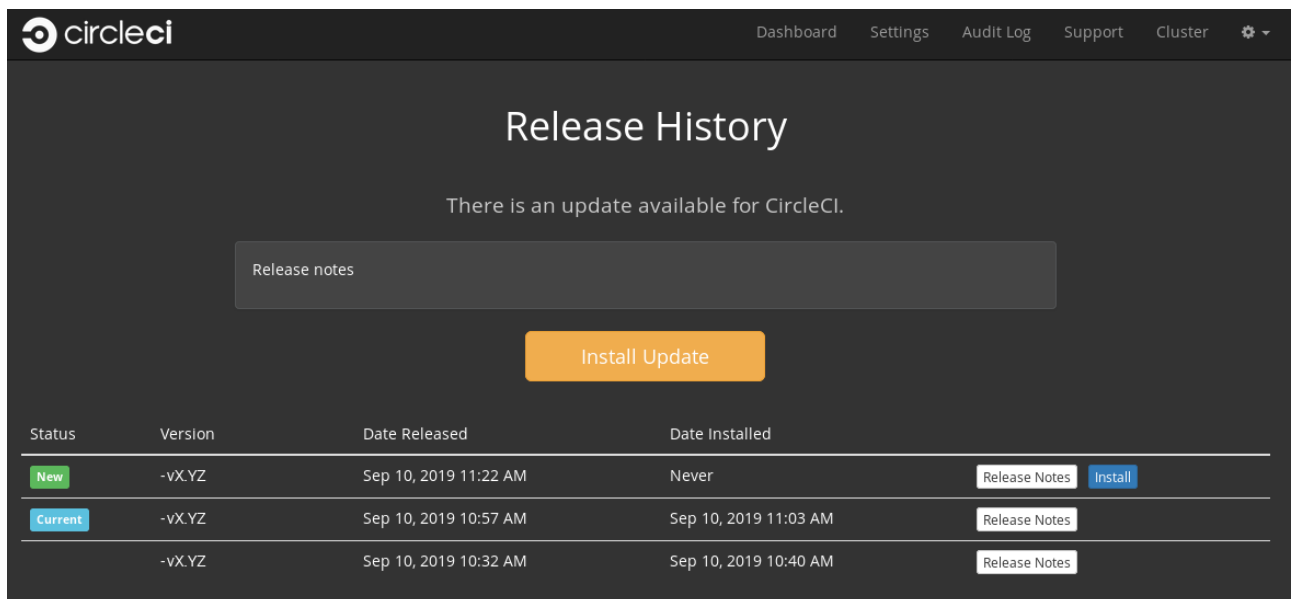


Figure 16. View Available Releases

The install process may take several minutes and the install status will be displayed both on the Releases

page and the main Dashboard.

3. Once the installation is finished, navigate to the Dashboard to start your installation - Note the middle box on the Dashboard will read "CircleCI is up to date" when you are running the latest version.