



Overview

An overview of the architecture, services, and new features of CircleCI Server v3.4.8

docs@circleci.com

Version 3.4.8, 02/02/2024: FINAL

CircleCI Server v3.x Overview	1
Introduction	1
Services	2
Platforms	7
CircleCI Server v3.x What's New	9
Release 3.4.8	9
Upgrade notes	9
Release 3.4.7	9
Upgrade notes	9
Changelog	9
Release 3.4.6	9
Upgrade notes	9
Changelog	9
Release 3.4.5	10
Upgrade notes	10
Changelog	10
Release 3.4.4	10
Upgrade notes	10
Changelog	10
Release 3.4.3	10
Upgrade notes	10
Changelog	10
Release 3.4.2	10
Upgrade notes	10
Changelog	11
Release 3.4.1	11
Upgrade notes	11
Changelog	11
Release 3.4.0	11
Upgrade notes	11
New Features	11
Changelog	11
Release 3.3.1	12
Changelog	12
Release 3.3.0	12
New features	12
Changelog	12
Release 3.2.2	12
Upgrade notes	12
Changelog	12
Release 3.2.1	12

Upgrade notes.....	12
Changelog	13
Release 3.2.0	13
Upgrade notes.....	13
PostgreSQL pod stuck in pending	13
Traefik pod fails to schedule	14
New features.....	14
Changelog	15
Release 3.1.0	15
Upgrade notes.....	15
New features.....	15
Changelog	16
Release 3.0.2	16
Changelog	16
Release 3.0.1	16
Changelog	16
CircleCI Server v3.x FAQ	17
Does Server 3.0 have a data retention policy?	17
What control is granted over Nomad certificates?	17
Is it possible to change or disable the polling time which checks for health status?	17
What do the Application Status options in KOTS mean?	17

CircleCI Server v3.x Overview

Introduction

CircleCI server is an on-premises CI/CD platform for enterprise customers who have compliance or security needs that require them to operate within their firewall, in a private cloud, or in a data center.

Server provides the same features as CircleCI's cloud offering, but operates within your Kubernetes cluster.

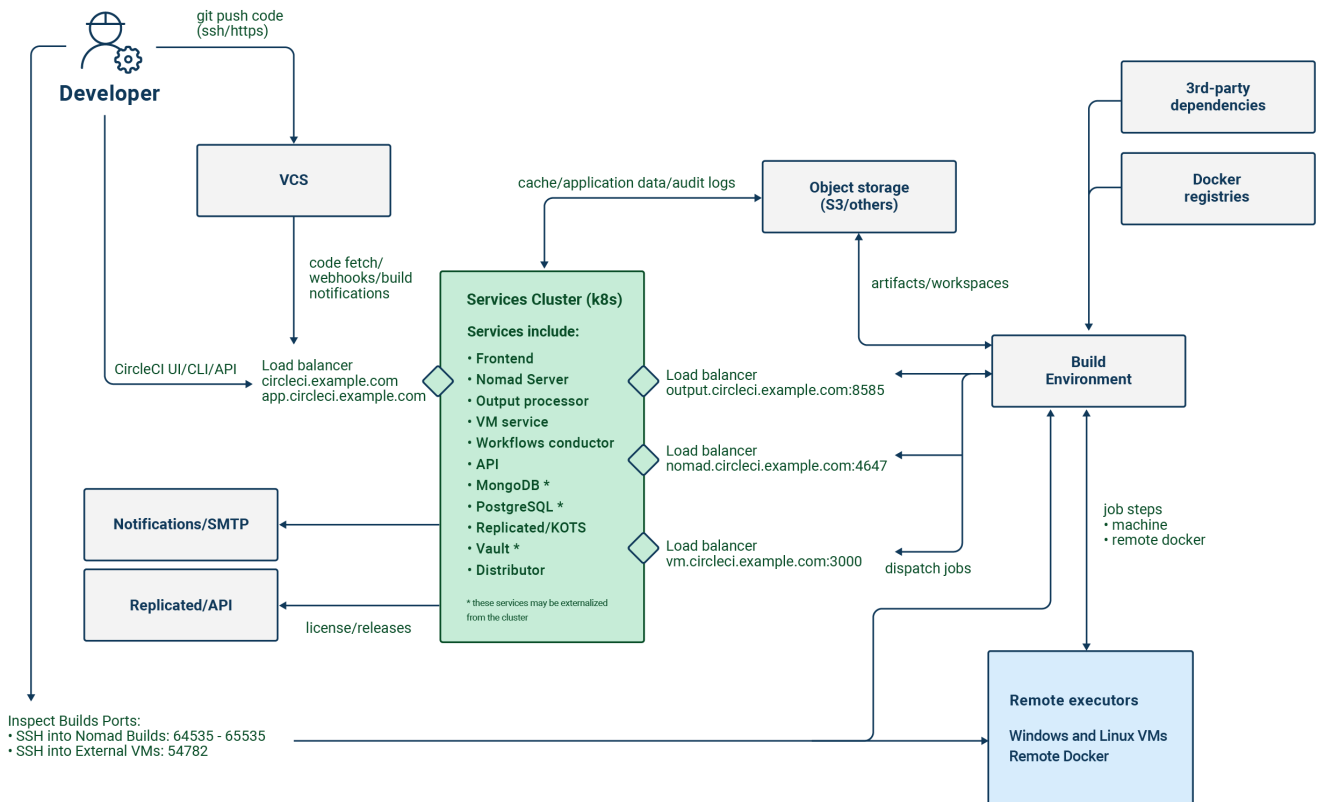


Figure 1. CircleCI Server v3.x Architecture

The CircleCI server application exposes four services, using load balancers. Three of these load balancers are VPC (Virtual Private Cloud)-internal for connecting to the Nomad cluster and virtual machines. If required, the front-end load balancer can be made private, separating it from the public internet. For further information see the [Load Balancers](#) doc.

Load Balancer	Type	Ports	Description
Frontend GUI Proxy & API	External	80 and 443	Exposes the web application.
Nomad Control Plane	Internal	4647	Exposes an RPC protocol for Nomad runners.
Output Processor	Internal	8585	Ingests output from Nomad runners.

Load Balancer	Type	Ports	Description
VM Service	Internal	3000	Provisions virtual machines.

The application exposes a number of external ports. These ports are used for various functions as defined in the table below.

Port number	Protocol	Direction	Source / Destination	Use	Notes
80	TCP	Inbound	End users	HTTP web app traffic	
443	TCP	Inbound	End users	HTTP web app traffic	
8800	TCP	Inbound	Administrators	Admin console	
22	TCP	Inbound	Administrators	SSH	Only required for the bastion host.
64535-65535	TCP	Inbound		SSH into builds	Only required for the Nomad clients.

CircleCI server schedules CI jobs using the [Nomad](#) scheduler. The Nomad control plane runs inside of Kubernetes, while the Nomad clients, which are responsible for running scheduled CircleCI jobs, are provisioned outside the cluster. CircleCI server can run Docker jobs on the Nomad clients themselves or in a dedicated virtual machine (VM).

Job artifacts and outputs are sent directly from jobs in Nomad to object storage (S3, GCS, or other supported options). Audit logs and other items from the application are also stored in object storage, so both the Kubernetes cluster and the Nomad clients need access to object storage.

Services

CircleCI server 3.0 consists of the following services. Find their descriptions and failure implications below:

Service	Component	Description	What happens if it fails?	Notes
api-service	App Core	Provides a GraphQL API that provides data to render the web frontend.	Many parts of the UI (such as Contexts) will fail completely.	

Service	Component	Description	What happens if it fails?	Notes
audit-log-service	App Core	Persists audit log events to blob storage for long-term storage.	Some events may not be recorded.	
builds-service	App Core	Ingests from www-api and sends to plans-service, workflows-conductor, and to orbs-service.		
circleci-mongodb	Execution	Primary datastore		
circleci-postgres	Data storage for microservices			
circleci-rabbitmq	Pipelines and Execution	Queuing for workflow messaging, test results, usage, crons, output, notifications, and scheduler.		
circleci-redis	Execution	Caches data that will not be stored permanently (such as build logs), for request caching, and for rate limit calculations.	A failed cache can result in rate limiting from the VCS if too many calls are made to it.	
circleci-teleggraf		Telegraf collects statsd metrics. All white-boxed metrics in our services publish statsd metrics that are sent to telegraf, but can also be configured to be exported to other places (such as Datadog or Prometheus).		

Service	Component	Description	What happens if it fails?	Notes
circleci-vault		HashiCorp Vault to run encryption and decryption as a service for secrets.		
config				
contexts-service	App Core	Stores and provides encrypted contexts.	All builds using Contexts will fail.	
cron-service	Pipelines	Triggers scheduled workflows.	Scheduled workflows will not run.	
dispatcher	Execution	Split jobs into tasks and send them to scheduler to run.	No jobs will be sent to Nomad. The run queue will increase in size, but there should be no meaningful loss of data.	
domain-service	App Core	Stores and provides information about our domain model. Works with permissions and API.	Workflows will fail to start and some REST API calls may fail, causing 500 errors in the CircleCI UI. If LDAP authentication is in use, all logins will fail.	
exim		Will be removed in GA, but users can provide mail submission credentials to an existing MTA.	No email notifications will be sent.	

Service	Component	Description	What happens if it fails?	Notes
frontend	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.	Rate limit of 150 requests per second with a single user instantaneous limit of 300 requests.
inject-bottoken		A Kubernetes job that inserts a "bot token" into MongoDB. Bot tokens are authorization interservice communication.		Mainly for www-api
kotsadm-kots	Licensing	The main KOTS application. Runs the KOTS Admin Console, where upgrades and server configuration take place.	No upgrades or configuration possible for server. No admin console available.	
kotsadm-migrations	Licensing	Performs database migrations to handle updates of Kotsadm.		
kotsadm-minio	Licensing	Object storage for KOTS licensing.		
kotsadm-postgres	Licensing	Database for KOTS licensing.		
legacy-notifier	App Core	Handles notifications to external services (for example, Slack or email).		
prometheus	Server	Used for metrics.		

Service	Component	Description	What happens if it fails?	Notes
orb-service	Pipelines	Handles communication between orb registry and config.		
output-processor	Execution	Receives job output and status updates and writes them to MongoDB. Also provides an API to running jobs to access caches, workspaces, store caches, workspaces, artifacts, & test results.		
permissions-service	App Core	Provides the CircleCI permissions interface.	Workflows will fail to start and some REST API calls may fail, causing 500 errors in the UI.	
scheduler	Execution	Runs tasks sent to it. Works with Nomad server.	No jobs will be sent to Nomad. The run queue will increase in size, but there should be no meaningful loss of data.	
slanger	server	Provides real-time events to the CircleCI app.	Live UI updates will stop, but hard refreshes will still work.	
test-results	Execution	Parses test result files and stores data.	There will be no test failure or timing data for jobs, but this will be back-filled once the service is restarted.	

Service	Component	Description	What happens if it fails?	Notes
vm-gc	Compute Management	Periodically checks for stale machine and remote Docker instances and requests that vm-service remove them.	Old vm-service instances might not be destroyed until this service is restarted.	
vm-scaler	Machine	Periodically requests that vm-service provision more instances for running machine and remote Docker jobs.	VM instances for machine and Remote Docker might not be provisioned, causing you to run out of capacity to run jobs with these executors.	Different overlay for EKS versus GKE.
vm-service	Machine	Inventory of available vm-service instances, and provisioning of new instances.	Jobs that use machine or remote Docker will fail.	
workflows-conductor-event-consumer	Pipelines	Takes in information from VCS to kick off pipelines.	New Pipelines will not be kicked off when there are changes in the VCS.	
workflows-conductor-grpc-handler	Pipelines	Helps translate the information through gRPC.		
web-ui-*	Frontend	Micro Front End (MFE) services used to render the frontend web application GUI.	The respective services page will fail to load. Example: A web-ui-server-admin failure means the server Admin page will fail to load.	The MFEs are used to render the web application located at app.<my domain here>

Platforms

CircleCI server is designed to deploy within a Kubernetes cluster. Virtual machine service (VM Service) is able to leverage unique EKS or GKE offerings to dynamically create VM images.

If installing outside of EKS or GKE, additional work is required to access some of the same machine build features. Setting up CircleCI runners gives you access to the same feature set as VM service across a much wider range of OSs and machine types (MacOS and much more).

We do our best to support a wide range of platforms for installation. We use environment-agnostic solutions wherever possible. However, we do not test all platforms and options. For that reason, we provide a list of tested environments, which we will continue to expand. We will be adding OpenShift to our list of regularly tested and supported platforms.

Environment	Status	Notes
EKS	Tested	
GKE	Tested	
Azure	Untested	Should work with Minio Azure gateway and Runner.
Digital Ocean	Untested	Should work with Minio Digital Ocean gateway and Runner.
OpenShift	Untested	Known to not work.
Rancher	Untested	Should work with Minio and Runner.

CircleCI Server v3.x What's New

Server 3.x is now generally available. The newest version of server offers the ability to scale under heavy workloads, all within your own Kubernetes cluster and private network, while still enjoying the full CircleCI cloud experience.

Server 3.x includes the latest CircleCI features, such as orbs, scheduled workflows, matrix jobs, and more. For existing customers interested in migrating from 2.19 to 3.x, contact your customer success manager. Server 3.x will receive monthly patch releases and quarterly feature releases.

Release 3.4.8

Upgrade notes

Before upgrading to v3.4.x, delete the deployment named `circleci-server-kube-state-metrics`:

1. Run `kubectl delete deployment/circleci-server-kube-state-metrics --cascade=orphan`
2. Deploy the app from the KOTS Admin Console == Changelog

For full details of this release see the [changelog](#).

Release 3.4.7

Upgrade notes

Before upgrading to v3.4.x, delete the deployment named `circleci-server-kube-state-metrics`:

1. Run `kubectl delete deployment/circleci-server-kube-state-metrics --cascade=orphan`
2. Deploy the app from the KOTS Admin Console

Changelog

For full details of this release see the [changelog](#).

Release 3.4.6

Upgrade notes

Before upgrading to v3.4.x, delete the deployment named `circleci-server-kube-state-metrics`:

1. Run `kubectl delete deployment/circleci-server-kube-state-metrics --cascade=orphan`
2. Deploy the app from the KOTS Admin Console

Changelog

For full details of this release see the [changelog](#).

Release 3.4.5

Upgrade notes

Before upgrading to v3.4.x, delete the deployment named `circleci-server-kube-state-metrics`:

1. Run `kubectl delete deployment/circleci-server-kube-state-metrics --cascade=orphan`
2. Deploy the app from the KOTS Admin Console

Changelog

For full details of this release see the [changelog](#).

Release 3.4.4

Upgrade notes

Before upgrading to v3.4.x, delete the deployment named `circleci-server-kube-state-metrics`:

1. Run `kubectl delete deployment/circleci-server-kube-state-metrics --cascade=orphan`
2. Deploy the app from the KOTS Admin Console

Changelog

For full details of this release see the [changelog](#).

Release 3.4.3

Upgrade notes

Before upgrading to v3.4.x, delete the deployment named `circleci-server-kube-state-metrics`:

1. Run `kubectl delete deployment/circleci-server-kube-state-metrics --cascade=orphan`
2. Deploy the app from the KOTS Admin Console

Changelog

For full details of this release see the [changelog](#).

Release 3.4.2

Upgrade notes

Before upgrading to v3.4.x, delete the deployment named `circleci-server-kube-state-metrics`:

1. Run `kubectl delete deployment/circleci-server-kube-state-metrics --cascade=orphan`

2. Deploy the app from the KOTS Admin Console

Changelog

For full details of this release see the [changelog](#).

Release 3.4.1

Upgrade notes

Before upgrading to v3.4.x, delete the deployment named `circleci-server-kube-state-metrics`:

1. Run `kubectl delete deployment/circleci-server-kube-state-metrics --cascade=orphan`
2. Deploy the app from the KOTS Admin Console

Changelog

For full details of this release see the [changelog](#).

Release 3.4.0

Upgrade notes

Before upgrading to v3.4.x, delete the deployment named `circleci-server-kube-state-metrics`:

1. Run `kubectl delete deployment/circleci-server-kube-state-metrics --cascade=orphan`
2. Deploy the app from the KOTS Admin Console

New Features

- Android images for the machine executor are now available. For more information, see the [Using Android Images with the Machine Executor](#) guide.
- The CircleCI Build Agent can now be pulled from a custom Docker registry if required. For more information contact customer support.
- You can now use Amazon Certificate Manager (ACM) to manage your TLS certificates. For more information see the [Server v3.x Installation Guide](#).
- Kubernetes v1.16 - v1.23 are now supported.
- Workload Identities for installations on GCP are now supported. Workload Identities can be used as an alternative to static credentials. For more information see the [Server v3.x Installation Guide](#)

Changelog

For full details of this release see the [changelog](#).

Release 3.3.1

Changelog

For full details of this release see the [changelog](#).

Release 3.3.0

New features

- Nomad Autoscaler can now be used to scale Nomad clients. For more information see the [execution environments installation docs](#).
- [Webhooks](#) are now available.
- The Insights dashboard is now available.
- IRSA (AWS) can now be used as an alternative to keys for object storage authentication.
- The email address from which build notifications are sent is now configurable from the KOTS Admin Console.
- We have replaced [Traefik](#) with [Kong](#) as our reverse proxy. However, in order to minimize disruption when upgrading, we chose not to rename the service used by kong. Therefore, you will see a service named `circleci-server-traefik`, however, this service is actually for Kong.

Changelog

For full details of this release see the [changelog](#).

Release 3.2.2

Upgrade notes

- The [rerun workflow](#) endpoint now returns workflow ID rather than the message `accepted`.

Changelog

For full details of this release see the [changelog](#).

Release 3.2.1

Upgrade notes

From the KOTS Admin Console, select **Version History** from the menu bar and click **Deploy** for server v3.2.0.

See [Upgrade notes](#) before upgrading from v3.1.x to v3.2.x.

Changelog

For full details of this release see the [changelog](#).

Release 3.2.0

Upgrade notes

From the KOTS Admin Console, select **Version History** from the menu bar and click **Deploy** for server v3.2.0.

When upgrading from server 3.1.x to 3.2, there will be some downtime due to a change to the PostgreSQL pod. There are two issues you could run into with this update, which are covered in the following sections.

PostgreSQL pod stuck in **pending**

If you find that the PostgreSQL pod is stuck in a **pending** state after upgrading, scale down the pods to 0 and then scale up again by following the steps below.

To check if your PostgreSQL pod is stuck in **pending**, use the following command:

```
$ kubectl get pod -l app.kubernetes.io/name=postgresql
NAME          READY   STATUS    RESTARTS   AGE
postgresql-0  1/1    Pending   0           3m
```

The following command will scale down pods to 0 and terminate the application pods without any data loss:

```
kubectl scale deployment -l layer=application --replicas 0
```

Once all the application-layer pods have finished terminating, do **one** of the following:

- **either** redeploy the update from the KOTS Admin Console
- **or** run the following two commands to redeploy the pods and return server to a functional state:

```
kubectl scale deployment -l layer=application --replicas 1
```

Then scale **output-processor** up with the following command:

```
kubectl scale deployment output-processor --replicas 2
```


Traefik pod fails to schedule

If you find that there are two Traefik pods after upgrading, you need to locate the older pod and remove it to allow the new pod to schedule correctly.

To see the status of your Traefik pod, use the following command:

```
$ kubectl get pod -l app=traefik
```

NAME	READY	STATUS	RESTARTS	AGE
circleci-server-traefik-9d6b86fd8-f7n2x	1/1	Running	0	24d
circleci-server-traefik-cf7d4d7f6-6mb5g	1/1	Error	0	3m

Remove the older Traefik pod with the following command:

```
kubectl delete pod circleci-server-traefik-<older pod hash>
```

The new Traefik pod will then start to schedule correctly.

New features

- Customers who require a fully private installation can now access a setting in the KOTS Admin Console to ensure public IPs are not assigned to VMs. Note that with this non-public IP setting enabled, a workaround will be needed if SSH access to running jobs is required, for example, by using a VPN into your VPC.
- Customers who manage outbound traffic through a proxy can now configure proxy settings through the KOTS Admin Console. Please see our documentation for specifics on [proxy support for server](#).
- We have expanded the machine execution environment options available to include additional resource classes, sizes, and executors. You now have access to Arm (medium, large), Linux (medium, large, X large, and XX large), and Windows (medium, large, XX large) resource classes.
- The [insights API](#) is now available to all server customers. Leverage build and other data to better understand the performance of teams and the health of your build and testing efforts.
- We have revamped the admin UI, and updated our installation instructions, making it easier to set up and manage server.
- You can now supply a custom Linux AMI for VM service.
- SSL termination can now be disabled. If you have put server login behind a firewall, this will enable SSL termination at the firewall.
- You can now control the size of persistent volumes. For larger customers, the initial persistent volume size was too small, by default. You can now set this at install time, providing an easier migration for those customers that require it. For further information see the [Internal Database Volume Expansion doc](#).
- We have added an auto-scaling example to the [nomad client terraform](#).
- You can now choose to serve 'unsafe' build artifacts. Previously this option was hidden,

meaning potentially unsafe artifacts were rendered as plain text. For more information see the [Build Artifacts doc](#).

Changelog

For full details of this release see the [changelog](#).

Release 3.1.0

Upgrade notes



With this release, the `frontend-external` load balancer has been removed. The `traefik` load balancer now handles all incoming traffic. When updating from a previous server 3.x version, you need to update the DNS record that was pointing to the `frontend-external` load balancer and point it to the `circleci-server-traefik` load balancer instead. Remember, you can retrieve the external IP address or DNS name of your Traefik load balancer by typing `kubectl get svc/circleci-server-traefik` in a terminal that has access to the cluster.

To update your DNS record and upgrade your server installation, follow these steps:

1. Retrieve the external IP or DNS name for the Traefik load balancer as described, or by looking the DNS A record for `app.<your domain name>` - this should already point to your Traefik load balancer.
2. Locate the DNS A record that points to the domain name of your server installation (not the one pointing to the `app.` subdomain).
3. Edit the A record so that it points to the Traefik load balancer, the same as the `app.` subdomain record. Your changes might need a couple of minutes to take effect, depending on your DNS service.

Next, from the KOTS Admin Console, select **Version History** from the menu bar and click **Deploy** for server v3.1.0.

New features

- The option to use only private load balancers has been introduced for customers who want a fully private installation. For more information, see the [Load Balancers](#) guide.
- Server 3.x hosts build artifacts, test results, and other state in object storage. We support any S3-compatible storage and Google Cloud Storage. For more information, see the [Installation guide](#).
- Dynamic config via setup workflows is now available on server installations. For more information, see our [blog post](#) and the [Dynamic Configuration docs page](#).
- Runner is now available on server. For further information, including installation steps, see the [Runner docs](#). Runner allows the use of the macOS executor in server installations and VM service functionality for customers with server installed in a private data centre.
- The frontend load balancer from v3.0 has been removed and replaced with an Ingress resource

and the Traefik Ingress controller. This is a breaking change requiring you to reconfigure your DNS. See the [What's New in server docs](#) for further information and guidance.

- The following services can now be externalized. For setup information, see the [server v3.x installation guide](#):
 - Postgres
 - MongoDB
 - Vault
- Backup and restore functionality is now available. For more information see the [Backup and Restore](#) guide.
- Server now supports the 2XL resource class. The Nomad cluster needs to be large enough to account for larger resource classes.
- The lifecycle of build artifacts and test results can now be configured from the KOTS Admin Console under **Storage Object Expiry**, including the option to disable the expiration and retain artifacts and test results indefinitely.

Changelog

For full details of this release see the [changelog](#).

Release 3.0.2

Changelog

For full details of this release see the [changelog](#).

Release 3.0.1

Changelog

For full details of this release see the [changelog](#).

CircleCI Server v3.x FAQ

Does Server 3.0 have a data retention policy?

Data used to power [Insights](#) and data returned by the [Insights APIs](#) when using Server 3.0 are subject to the retention policy stated on the API's documentation. All other data used for Server 3.0 does not have a defined retention policy and is stored indefinitely.

What control is granted over Nomad certificates?

Full control of the certificates, all the way down to mTLS for Nomad.

Is it possible to change or disable the polling time which checks for health status?

No, this is not customizable.

What do the Application Status options in KOTS mean?

Please see the [KOTS documentation](#).