CircleCI Server v2.17 Operations Guide

May 13th, 2019

Contents

Overview

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.

CircleCI Server v2.17 uses the CircleCI 2.0 architecture.

Build Environments

CircleCI 2.0 uses Nomad as the primary job scheduler. Refer to the Introduction to Nomad Cluster Operation 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 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:

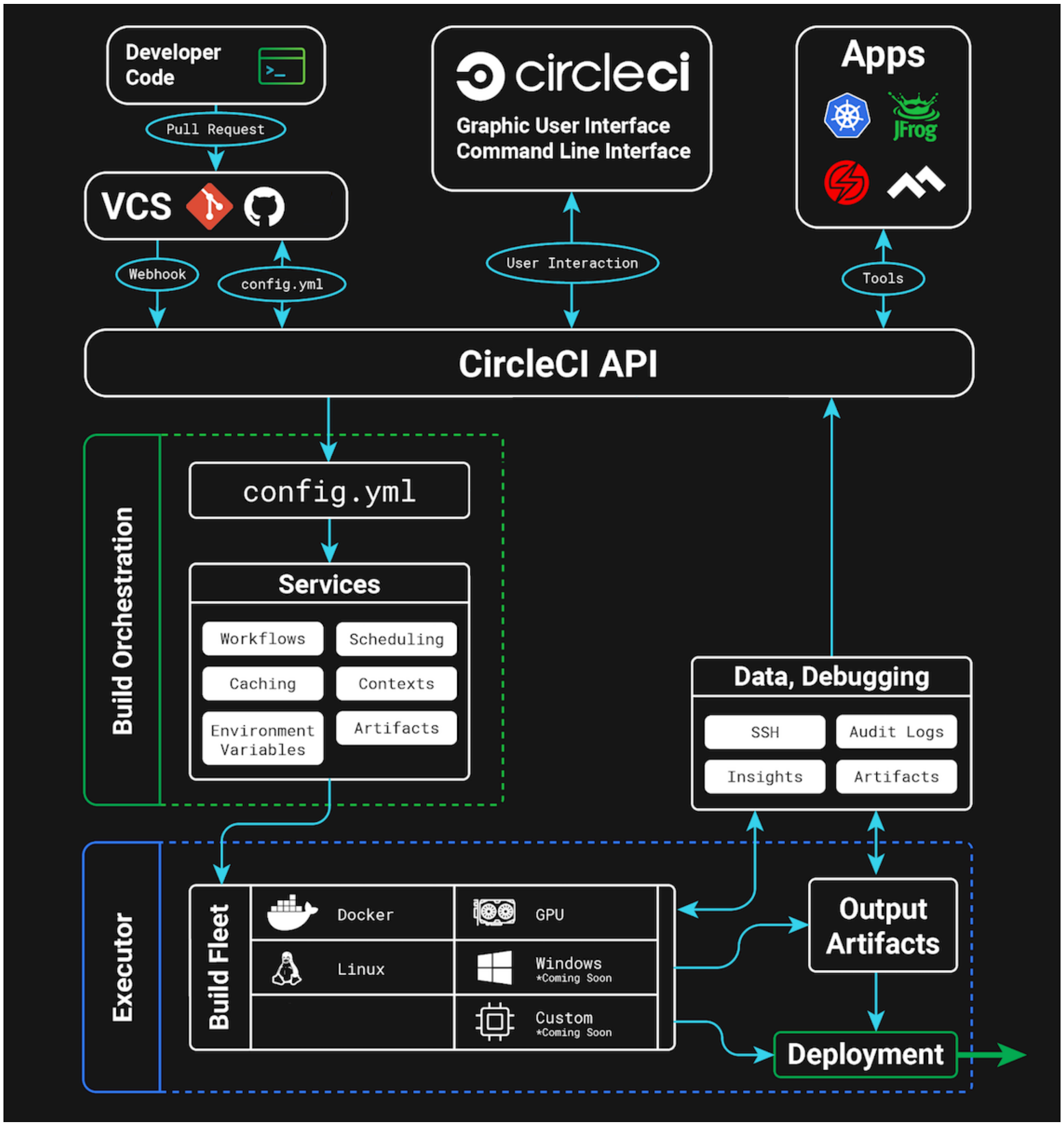

Figure 1: CircleCI Services Architecture

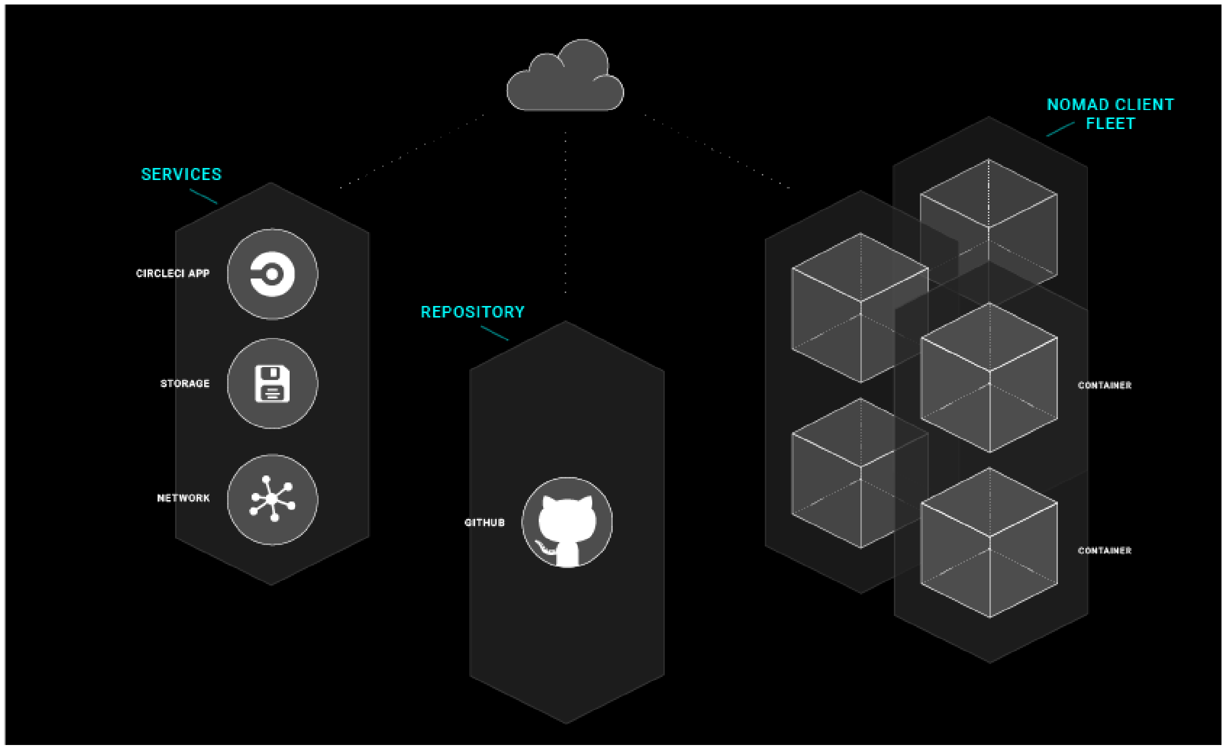

Figure 2: A Diagram of the CircleCI Architecture

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 Advanced System Monitoring.

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.

Note: 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. **Note:** 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

Server Ports

This chapter provides System Administrators with a complete list of ports for the machines in their CircleCI 2.0 installation:

Machine type	Port number	Protocol	Direction	Source / destination	Use	Notes
Services Machine	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	
	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 externalised 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	

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

Machine type	Port number	Protocol	Direction	Source / destination	Use	Notes
	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 (whichever applies)	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	
	443	TCP	Bi-directional	Services Machine	Webhooks / API access	
PostgreSQL Servers	5432	TCP	Inbound	Services Machine	PostgreSQL database connection	Only if using externalised databases. Port is user-defined, assuming the default PostgreSQL port.

Machine type	Port number	Protocol	Direction	Source / destination	Use	Notes
	5432	TCP	Bi-directional	PostgreSQL Servers	PostgreSQL replication	Only if using externalised databases. Port is user-defined, assuming the default PostgreSQL port.
MongoDB Servers	27017	TCP	Inbound	Services Machine	MongoDB database connection	Only if using externalised databases. Port is user-defined, assuming the default MongoDB port.
	27017	TCP	Bi-directional	MongoDB Servers	MongoDB replication	Only if using externalised databases. Port is user-defined, assuming the default MongoDB port.
RabbitMQ Servers	5672	TCP	Inbound	Services Machine	RabbitMQ connection	Only if using externalised RabbitMQ
	5672	TCP	Bi-directional	RabbitMQ Servers	RabbitMQ mirroring	Only if using externalised RabbitMQ
Redis Servers	6379	TCP	Inbound	Services Machine	Redis connection	Only if using externalised Redis
	6379	TCP	Bi-directional	Redis Servers	Redis replication	Only if using externalised Redis and using Redis replication (optional)
Nomad Servers	4646	TCP	Inbound	Services Machine	Nomad Server connection	Only if using externalised Nomad Servers
	4647	TCP	Inbound	Services Machine	Nomad Server connection	Only if using externalised Nomad Servers

Machine type	Port number	Protocol	Direction	Source / destination	Use	Notes
	4648	TCP	Bi-directional	Nomad Servers	Nomad Servers internal communication	Only if using externalised Nomad Servers

Installing CircleCI 2.x with Static Scripts

This chapter provides a overview of installing CircleCI 2.x in a private cloud or datacenter by using static `init` scripts.

Limitations

This method of installation has the following limitations:

- It is not possible to use `machine` executors.
- It is not possible to use the Remote Docker Environment or Docker Layer Caching.
- There is no first-class high-availability option.

Build Environments

By default, the Nomad Client instances automatically provision containers according to the image configured for each job in your `.circleci/config.yml` file. CircleCI uses Nomad as the primary job scheduler in CircleCI 2.x. Refer to the Introduction to Nomad Cluster Operation to learn more about the job scheduler and how to perfom basic client and cluster operations.

Architecture

A CircleCI static installation consists of two primary components: Services and Nomad Clients. Services run on a single instance that is comprised of the core application, storage, and networking functionality. Any number of Nomad Clients execute jobs and communicate back to the Services machine. Both components must access an instance of GitHub or GitHub Enterprise on the network as illustrated in the following architecture diagram.

Services

The machine on which the Services instance runs should only be restarted gracefully and may be backed up using built-in VM snapshotting. **Note:** It is possible to configure external data storage with PostgreSQL and Mongo for high availability and then use standard tooling for database backups, see Adding External Database Hosts for High Availability. DNS resolution must point to the IP address of the machine on which the Services are installed. The following table describes the ports used for traffic on the Service instance:

Source	Ports	Use
End Users	80, 443, 7171, 8081	HTTP/HTTPS Traffic

Source	Ports	Use
Administrators	22	SSH
Administrators	8800	Admin Console
Nomad Clients	4647, 8585, 7171, 3001	Internal Communication
GitHub (Enterprise or .com)	80, 443	Incoming Webhooks

Nomad Clients

The Nomad Client instances run without storing state, enabling you to increase or decrease containers as needed. To ensure that there are enough client machines running to handle all of the builds, track the queued builds, and increase the client machines as needed to balance the load.

Each machine on which the Nomad Clients are installed reserves two CPUs 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 following table describes the ports used on the Nomad client instances:

Source	Ports	Use
End Users	64535-65535	SSH into builds feature
Administrators	80 or 443	CircleCI API Access (graceful shutdown, etc)
Administrators	22	SSH
Services VM	4647, 8585, 7171, 3001	Internal Communication

GitHub

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

Installation

The following sections describe the steps for installation of the Services VM and the Nomad cluster.

Prerequisites

Have the following available before beginning the installation procedure:

- A CircleCI License file (.rli). Contact CircleCI support if you need a license.
- A machine to run Ubuntu 14.04 with a minimum of at least 100 GB storage, 32 GB RAM, and 4 CPUs (8 CPUs preferred) for the Services VM.

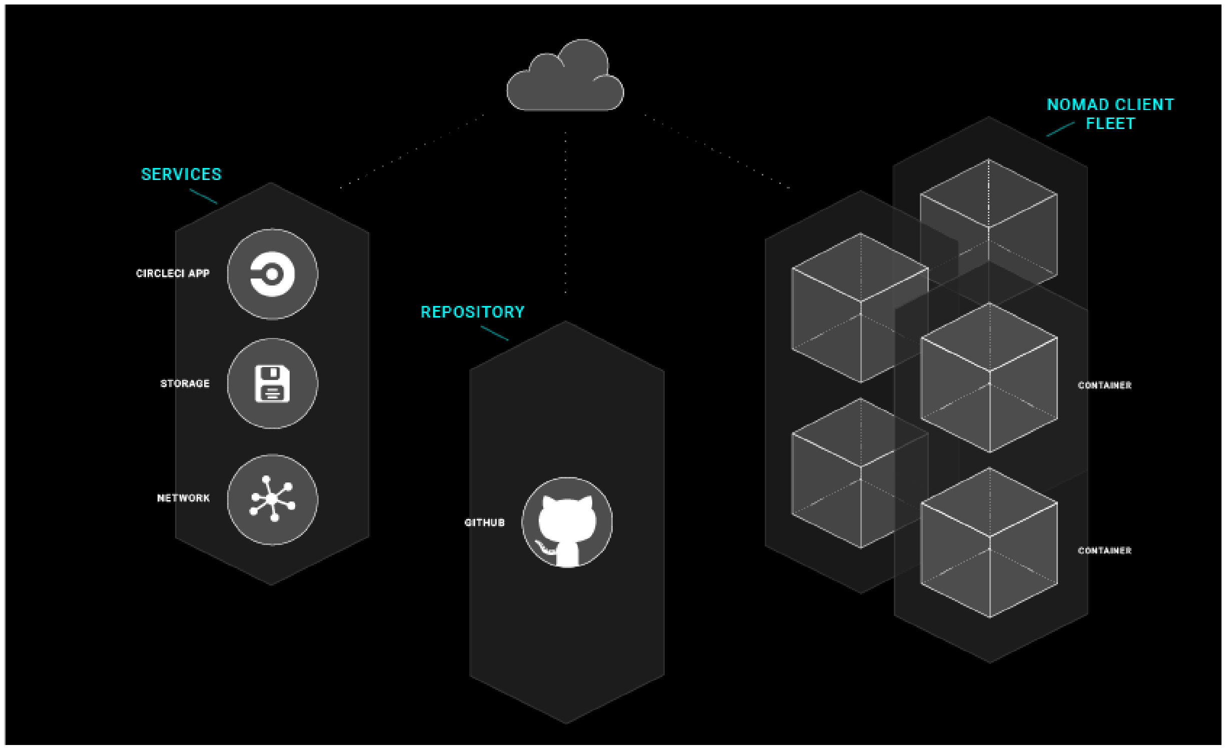

Figure 3: A Diagram of the CircleCI Architecture

- A cluster of machines running Ubuntu 14.04 with a minumum of 8 GB RAM and 4 CPUs each, as well as network access to any Docker registries that are required by your builds for the Nomad Client VMs.

Installing the Services Machine

1. Copy the Services init script to the Services VM machine.
2. Log in to the machine provisioned for the Services VM and run the `sudo su` command.
3. Run `./provision-services-ubuntu.sh` to start the script.
4. Go to the public IP of the host on port 8800 using HTTPS.
5. Enter your license.
6. Complete the Storage section. If you are not using a cloud service, then you will pick `None` (more information below).
7. Set the VM Provider to None.
8. Set 1.0 Builds to Off.
9. Set 2.0 Builds to Clustered. **Note:** If this option is disabled, contact your service representative.

Installing the Nomad Clients

1. Copy the Client init script to the Nomad Server machine.
2. Log in to the machine provisioned for the Nomad Server and run the `sudo su` command.

3. To start the script, run `./provision-nomad-client-ubuntu.sh` with the `NOMAD_SERVER_ADDRESS` environment variable set to the routable IP of the Services machine (for example, if you are provisioning your nomad client in the same machine as services maschine, use `NOMAD_SERVER_ADDRESS=0.0.0.0 ./provision-nomad-client-ubuntu.sh`).

Storage

The `None` storage driver saves all of your CircleCI data locally. This means that artifacts, test results, and action logs will be saved locally at `/data/circle/storage-fileserver`. It is best practice to mount an external volume and create a symbolic link between the two when using this storage option. **Note:** Data may only be transferred as quickly as the external volume will allow, so SSDs are best practice.

Troubleshooting

This section includes some possible resolutions for common issues that may be encountered during system setup and installation.

- Symptom: Jobs stay in `queued` status until they fail and never successfully run.
  - Check port 8585 if the nomad client logs contain the following type of error message:
    - {"error":"rpc error: code = Unavailable desc = grpc: the connection is unavailable","level":"warning","msg":"error fetching config, retrying","time":"2018-04-17T18:47:01Z"}

Troubleshooting

This chapter answers frequently asked questions and provides installation troubleshooting tips.

FAQ

Can I move or change my GitHub Enterprise URL without downtime?

No, because of the nature of CircleCI integration with GitHub authentication, you should not change the domain of your GHE instance after CircleCI is in production. Redeploying GitHub without will result in a corrupted CircleCI instance. Contact support if you plan to move your GitHub instance.

Can I monitor available build containers?

Yes, refer to the Introduction to Nomad Cluster Operation document for details. Refer to the Administrative Variables, Monitoring, and Logging section for how to enable additional container monitoring for AWS.

How do I provision admin users?

The first user who logs in to the CircleCI application will automatically be designated an admin user. Options for designating additional admin users are found under the Users page in the Admin section at `https://[domain-to-your-installation]/admin/users`.

How can I gracefully shutdown Nomad Clients?

Refer to the Introduction to Nomad Cluster Operation chapter for details.

Why is Test GitHub Authentication failing?

This means that the GitHub Enterprise server is not returning the intermediate SSL certificates. Check your GitHub Enterprise instance with https://www.ssllabs.com/ssltest/analyze.html - it may report some missing intermediate certs. You can use commands like `openssl` to get the full certificate chain for your server.

In some cases authentication fails when returning to the configuration page after it was successfully set up once. This is because the secret is encrypted, so when returning checking it will fail.

How can I use HTTPS to access CircleCI?

While CircleCI creates a self-signed cert when starting up, that certificate only applies to the management console and not the CircleCI product itself. If you want to use HTTPS, you'll have to provide certificates to use under the `Privacy` section of the settings in the management console.

Why doesn't terraform destroy every resource?

CircleCI sets the services box to have termination protection in AWS and also writes to an s3 bucket. If you want terraform to destroy every resource, you'll have to either manually delete the instance, or turn off termination protection in the `circleci.tf` file. You'll also need to empty the s3 bucket that was created as part of the terraform install.

Do the Nomad Clients store any state?

They can be torn down without worry as they don't persist any data.

How do I verify TLS settings are failing?

Make sure that your keys are in unencrypted PEM format, and that the certificate includes the entire chain of trust as follows:

```
-----BEGIN CERTIFICATE-----
your_domain_name.crt
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
intermediate 1
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
intermediate 2
-----END CERTIFICATE-----
...
```

How do I debug the Management Console (Replicated)?

The CircleCI management console is powered by Replicated. If you are experiencing any issues with the Management Console, here are a few ways to debug it:

1. Check you have Replicated installed

First, make sure you have the CLI tool for Replicated installed by running the following:

```
replicated –version
```

2. Restart Replicated and the CircleCI app

Try restarting Replicated services. You can do this by running the following commands on the service box, for Ubuntu 14.04:

```
sudo service replicated-ui restart
sudo service replicated restart
sudo service replicated-operator restart
```

For Ubuntu 16.04, run the following commands:

```
sudo systemctl restart replicated-ui
sudo systemctl restart replicated
sudo systemctl restart replicated-operator
```

Then try restarting the CircleCi app: go to your services box admin (for example, `https://<your-circleci-hostname>.com:8800`) and try restarting with "Stop Now" and "Start Now".

3. Try to log into Replicated

Try logging in to Replicated. You can do this by running the following command on the service box. You will be asked to enter your password - the same one used to unlock the Management Console (i.e. `https://<your-circleci-hostname>.com:8800`).

```
replicated login
```

If you could login, then run the following command and send the output to us at support@circleci.com so we can help diagnose what is causing the problem you are experiencing.

```
sudo replicated apps
```

If you were seeing the following error: `request returned Unauthorized for API route` this could be because you are not logged into Replicated, so please check if you are still getting the error after a successful login.

4. Check Replicated logs

You can find Replicated logs on the Services machine under `/var/log/replicated`.

5. Check what Docker containers are currently running

Replicated starts many Docker containers to run CircleCI Server, so it can be useful to check what containers are running.

To check what containers are currently running, run `sudo docker ps` and you should see something similar to this output:

```
$ sudo docker ps
CONTAINER ID     IMAGE                                                        COMMAND              CREATED         STATUS          PORTS
eb2970306859        172.31.72.162:9874/circleci-api-service:0.1.6910-8b54ef9              "circleci-service-run"   26 hours
ago        Up 26 hours         0.0.0.0:32872->80/tcp, 0.0.0.0:32871->443/tcp, 0.0.0.0:8082->3000/tcp,
0.0.0.0:32870->6010/tcp, 0.0.0.0:32869->8585/tcp                                  api-service

01d26714f5f5        172.31.72.162:9874/circleci-workflows-conductor:0.1.38931-1a904bc8    "/service/docker-ent…"   26 hours
ago        Up 26 hours         0.0.0.0:9998->9998/tcp, 0.0.0.0:32868->80/tcp, 0.0.0.0:32867->443/tcp,
0.0.0.0:9999->3000/tcp, 0.0.0.0:32866->8585/tcp                                   workflows-conductor

0cc6e4248cfb        172.31.72.162:9874/circleci-permissions-service:0.1.1195-b617002      "/service/docker-ent…"   26 hours
ago        Up 26 hours         0.0.0.0:3013->3000/tcp
permissions-service

9e6efc98b7d6        172.31.72.162:9874/circleci-cron-service:0.1.680-1fcd8d2              "circleci-service-run"   26 hours
ago       Up 26 hours        0.0.0.0:4261->4261/tcp                                                                                         cron-service
8c40bd1cecf6        172.31.72.162:9874/circleci-federations-service:0.1.1134-72edcbc      "/service/docker-ent…"   26 hours
ago      Up 26 hours       0.0.0.0:3145->3145/tcp, 0.0.0.0:8010->8010/tcp, 0.0.0.0:8090->8090/tcp                                   federations-service
71c71941684f        172.31.72.162:9874/circleci-contexts-service:0.1.6073-5275cd5         "./docker-entrypoint…"   26 hours
ago      Up 26 hours       0.0.0.0:2718->2718/tcp, 0.0.0.0:3011->3011/tcp, 0.0.0.0:8091->8091/tcp                                     contexts-service
71ffeb230a90        172.31.72.162:9874/circleci-domain-service:0.1.4040-eb63b67           "/service/docker-ent…"   26 hours
ago       Up 26 hours        0.0.0.0:3014->3000/tcp                                                                                       domain-service
eb22d3c10dd8        172.31.72.162:9874/circleci-audit-log-service:0.1.587-fa47042         "circleci-service-run"   26 hours
ago       Up 26 hours                                                                                                                  audit-log-service
243d9082e35c        172.31.72.162:9874/circleci-frontend:0.1.203321-501fada               "/docker-entrypoint.…"   26 hours
ago       Up 26 hours        0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp, 0.0.0.0:4434->4434/tcp                                                     frontend
af34ca3346a7        172.31.72.162:9874/circleci-picard-dispatcher:0.1.10401-aa50e85       "circleci-service-run"   26 hours
ago       Up 26 hours                                                                                                                  picard-dispatcher
fb0ee1b02d48         172.31.72.162:9874/circleci-vm-service:0.1.1370-ad05648                "vm-service-service-…"   26 hours ago        Up 26 hours         0.0.0.0:3001-
>3000/tcp                                                                                                            vm-service
```

```
3708dc80c63e        172.31.72.162:9874/circleci-vm-scaler:0.1.1370-ad05648                "/scaler-entrypoint.…"   26 hours
ago       Up 26 hours        0.0.0.0:32865->5432/tcp                                                                               vm-scaler
77bc9d0b4ac9        172.31.72.162:9874/circleci-vm-gc:0.1.1370-ad05648                    "docker-entrypoint.s…"   26 hours
ago       Up 26 hours        0.0.0.0:32864->5432/tcp                                                                                   vm-gc
4b02f202a05d        172.31.72.162:9874/circleci-output-processing:0.1.10386-741e1d1       "output-processor-se…"   26 hours
ago       Up 26 hours        0.0.0.0:8585->8585/tcp, 0.0.0.0:32863->80/tcp, 0.0.0.0:32862->443/tcp                               picard-
output-processor
b8f982d32989        172.31.72.162:9874/circleci-frontend:0.1.203321-501fada               "/docker-entrypoint.…"   26 hours ago        Up 26 hours         0.0.0.0:32861-
>80/tcp, 0.0.0.0:32860->443/tcp, 0.0.0.0:32859->4434/tcp                                                                  dispatcher
601c363a0c38        172.31.72.162:9874/circleci-frontend:0.1.203321-501fada               "/docker-entrypoint.…"   26 hours
ago      Up 26 hours       0.0.0.0:32858->80/tcp, 0.0.0.0:32857->443/tcp, 0.0.0.0:32856->4434/tcp                          legacy-notifier
f2190c5f3aa9        172.31.72.162:9874/mongo:3.6.6-jessie                                 "/entrypoint.sh"         26 hours
ago       Up 26 hours        0.0.0.0:27017->27017/tcp                                                                              mongo
3cbbd959f42e        172.31.72.162:9874/telegraf:1.6.4                                     "/telegraf-entrypoin…"   26 hours
ago       Up 26 hours        0.0.0.0:8125->8125/udp, 0.0.0.0:32771->8092/udp, 0.0.0.0:32855->8094/tcp                           telegraf
15b090e8cc02        172.31.72.162:9874/circleci-schedulerer:0.1.10388-741e1d1             "circleci-service-run"   26 hours
ago       Up 26 hours                                                                                                   picard-scheduler
fb967bd3bca0        172.31.72.162:9874/circleci-server-nomad:0.5.6-5.1                    "/nomad-entrypoint.sh"   26 hours
ago       Up 26 hours        0.0.0.0:4646-4648->4646-4648/tcp                                                                      nomad
7e0743ee2bfc        172.31.72.162:9874/circleci-test-results:0.1.1136-b4d94f6             "circleci-service-run"   26 hours
ago       Up 26 hours        0.0.0.0:2719->2719/tcp, 0.0.0.0:3012->3012/tcp                                                 test-results
0a95802c87dc        172.31.72.162:9874/circleci-slanger:0.4.117-42f7e6c                   "/docker-entrypoint.…"   26 hours
ago       Up 26 hours        0.0.0.0:4567->4567/tcp, 0.0.0.0:8081->8080/tcp                                                      slanger
ca445870a057        172.31.72.162:9874/circleci-postgres-script-enhance:0.1.9-38edabf     "docker-entrypoint.s…"   26 hours
ago       Up 26 hours        0.0.0.0:5432->5432/tcp                                                                             postgres
a563a228a93a        172.31.72.162:9874/circleci-server-ready-agent:0.1.105-0193c73        "/server-ready-agent"    26 hours
ago       Up 26 hours        0.0.0.0:8099->8000/tcp                                                                          ready-agent
d6f9aaae5cf2        172.31.72.162:9874/circleci-server-usage-stats:0.1.122-70f28aa        "bash -c /src/entryp…"   26 hours
ago       Up 26 hours                                                                                                      usage-stats
086a53d9a1a5        registry.replicated.com/library/statsd-graphite:0.3.7                 "/usr/bin/supervisor…"   26 hours
ago      Up 26 hours       0.0.0.0:32851->2443/tcp, 0.0.0.0:32770->8125/udp                                          replicated-statsd
cc5e062844be        172.31.72.162:9874/circleci-shutdown-hook-poller:0.1.32-9c553b4       "/usr/local/bin/pyth…"   26 hours
ago       Up 26 hours                                                                                                   musing_volhard
9609f04c2203        172.31.72.162:9874/circleci-rabbitmq-delayed:3.6.6-management-12      "docker-entrypoint.s…"   26 hours
ago        Up 26 hours         0.0.0.0:5672->5672/tcp, 0.0.0.0:15672->15672/tcp, 0.0.0.0:32850->4369/tcp, 0.0.0.0:32849->5671/tcp, 0.0.0.0:32848->15671/tcp, 0.0.0.0:32847-
>25672/tcp   rabbitmq
2bc0cfe43639        172.31.72.162:9874/tutum-logrotate:latest                             "crond -f"               26 hours
ago       Up 26 hours                                                                                                    hardcore_cray
79aa857e23b4        172.31.72.162:9874/circleci-vault-cci:0.3.8-e2823f6                   "./docker-entrypoint…"   26 hours
ago       Up 26 hours        0.0.0.0:8200-8201->8200-8201/tcp                                                                 vault-cci
b3e317c9d62f        172.31.72.162:9874/redis:4.0.10                                       "docker-entrypoint.s…"   26 hours
ago       Up 26 hours        0.0.0.0:6379->6379/tcp                                                                                redis
f2d3f77891f0        172.31.72.162:9874/circleci-nomad-metrics:0.1.90-1448fa7              "/usr/local/bin/dock…"   26 hours
ago       Up 26 hours                                                                                                     nomad-metrics
1947a7038f24        172.31.72.162:9874/redis:4.0.10                                       "docker-entrypoint.s…"   26 hours
ago       Up 26 hours        0.0.0.0:32846->6379/tcp                                                                      slanger-redis
3899237a5782        172.31.72.162:9874/circleci-exim:0.2.54-697cd08                       "/docker-entrypoint.…"   26 hours
ago       Up 26 hours        0.0.0.0:2525->25/tcp                                                                                   exim
97ebdb831a7e        registry.replicated.com/library/retraced:1.2.2                        "/src/replicated-aud…"   26 hours
ago       Up 26 hours        3000/tcp                                                                                 retraced-processor
a0b806f3fad2        registry.replicated.com/library/retraced:1.2.2                        "/src/replicated-aud…"   26 hours
ago       Up 26 hours        172.17.0.1:32771->3000/tcp                                                                     retraced-api
19dec5045f6e        registry.replicated.com/library/retraced:1.2.2                        "/bin/sh -c '/usr/lo…"   26 hours
ago       Up 26 hours        3000/tcp                                                                                      retraced-cron
7b83a3a193da        registry.replicated.com/library/retraced-postgres:10.5-20181009       "docker-entrypoint.s…"   26 hours
ago       Up 26 hours        5432/tcp                                                                                  retraced-postgres
029e8f454890        registry.replicated.com/library/retraced-nsq:v1.0.0-compat-20180619   "/bin/sh -c nsqd"        26 hours
ago       Up 26 hours        4150-4151/tcp, 4160-4161/tcp, 4170-4171/tcp                                                   retraced-nsqd
500619f53e80        quay.io/replicated/replicated-operator:current                        "/usr/bin/replicated…"   26 hours
ago       Up 26 hours                                                                                                replicated-operator
e1c752b4bd6c        quay.io/replicated/replicated:current                                 "entrypoint.sh -d"       26 hours
ago       Up 26 hours        0.0.0.0:9874-9879->9874-9879/tcp                                                                replicated
1668846c1c7a        quay.io/replicated/replicated-ui:current                              "/usr/bin/replicated…"   26 hours
ago       Up 26 hours        0.0.0.0:8800->8800/tcp                                                                       replicated-ui
f958cf3e8762        registry.replicated.com/library/premkit:1.2.0                         "/usr/bin/premkit da…"   3 weeks
ago       Up 26 hours       80/tcp, 443/tcp, 2080/tcp, 0.0.0.0:9880->2443/tcp                                        replicated-premkit
```

Providing support@circleci.com with the output of `sudo docker ps` from the Services machine will help us diagnose the cause of your problem.