

# Anaconda Enterprise Notebook Runbook

---

Anaconda Enterprise Notebook is a Python data analysis environment from Continuum Analytics. Accessed through a browser, Anaconda Enterprise Notebook is a ready-to-use, powerful, fully-configured Python analytics environment. We believe that programmers, scientists, and analysts should spend their time analyzing data, not working to set up a system. Data should be shareable, and analysis should be repeatable. Reproducibility should extend beyond just code to include the runtime environment, configuration, and input data.

Anaconda Enterprise Notebook makes it easy to start your analysis immediately.

This runbook walks through the steps needed to install a basic Anaconda Enterprise Notebook system comprised of the front-end server, gateway, and two compute machines. The runbook is designed for two audiences: those who have direct access to the internet for installation and those where such access is not available or restricted for security reasons. For these restricted a.k.a. "Air Gap" environments, Continuum ships the entire Anaconda product suite on portable storage medium or as a downloadable TAR archive. Where necessary, additional instructions for Air Gap environments are noted. If you have any questions about the instructions, please contact your sales representative or Priority Support team, if applicable, for additional assistance.

**Wakari Server:** The administrative front-end to the system. This is where users login to the system, where user accounts are stored, and where admins can manage the system.

**Wakari Gateway:** The gateway is a reverse proxy that authenticates users and automatically directs them to the proper Wakari Compute machine for their project. Users will not notice this component as it automatically routes them. One could put a gateway in each datacenter in a tiered scale-out fashion.

**Wakari Compute nodes:** This is where projects are stored and run. Wakari Compute machines only need to be reachable by the Wakari Gateway, so they can be completely isolated by a firewall.

## Requirements

---

### Hardware Recommendations

#### Wakari Server

- 2+GB RAM
- 2+CPU cores
- 20GB storage

#### Wakari Gateway

- 2 GB RAM
- 2 CPU cores

#### Wakari Compute (N-machines)

Configure to meet the needs of the projects. At least:

- 2GB RAM
- 2 CPU cores

### OS Requirements

- RHEL/CentOS 6.7 on all nodes (Other operating systems are supported, however this document assumes RHEL or CentOS 6.7)
- **/opt/wakari:** Ability to install here and at least 5GB of storage.
- **/projects:** Size depends on number and size of projects. At least 20GB of storage.

**NOTE:** This directory needs the filesystem mounted with Posix ACL support (Posix.1e). Check with `mount` and

```
tune2fs -l /path/to/filesystem | grep options
```

## Software Prerequisites

### Wakari Server

- Mongo Version:  $\geq 2.6.8$  and  $< 3.0$
- Nginx version:  $\geq 1.4.0$
- ElasticSearch
- Oracle JRE 8

**NOTE:** For Air Gap installations, Oracle JRE must already be installed

### Wakari Compute

- git

## Security Requirements

- root or sudo access
- SELinux in Permissive mode - check with `getenforce`

## Network Requirements

### TCP Ports

- Server: 80
- Gateway: 8080
- Compute: 5002

## Other Requirements

Assuming the above requirements are met, there are no additional dependencies necessary for Wakari Enterprise.

## Air Gap vs. Regular Installation

As stated previously, this document contains installation instructions for two audiences: those with internet access on the destination server(s) and those who have no access to internet resources. Many of the steps below have two sections: **Air Gap Installation** and **Regular Installation**. Those without internet access should follow the **Air Gap Installation** instructions and those with internet access should follow **Regular Installation** instructions.

## Air Gap Media

This document assumes that the Air Gap media is located at `/installer` on the server where the software is being installed.

Air Gap media contents:

```
/installer
mongodb-org-tools-2.6.8-1.x86_64.rpm
mongodb-org-shell-2.6.8-1.x86_64.rpm
mongodb-org-server-2.6.8-1.x86_64.rpm
mongodb-org-mongos-2.6.8-1.x86_64.rpm
mongodb-org-2.6.8-1.x86_64.rpm
wakari-compute-0.10.0-Linux-x86_64.sh
wakari-server-0.10.0-Linux-x86_64.sh
wakari-gateway-0.10.0-Linux-x86_64.sh
wakari-publisher-0.10.0-Linux-x86_64.sh
nginx-1.6.2-1.el6ngx.x86_64.rpm
elasticsearch-1.7.2.noarch.rpm
jre-8u65-linux-x64.rpm
```

## Download the Installers

Download the installers and copy them to the corresponding servers. The Publisher should be installed on the Wakari Server machine.

- **Air Gap Installation:** Copy installers from the Air Gap media
- **Regular Installation:**

```
RPM_CDN="https://820451f3d8380952ce65-4cc6343b423784e82fd202bb87cf87cf.ssl.cf1.rackcdn.com"
curl -O $RPM_CDN/wakari-server-0.10.0-Linux-x86_64.sh
curl -O $RPM_CDN/wakari-gateway-0.10.0-Linux-x86_64.sh
curl -O $RPM_CDN/wakari-compute-0.10.0-Linux-x86_64.sh
curl -O $RPM_CDN/wakari-publisher-0.10.0-Linux-x86_64.sh
```

## Gather IP addresses or FQDNs

Wakari is very sensitive to the IP address or domain name used to connect to the Server and Gateway components. If users will be using the domain name, you should install the components using the domain name instead of the IP addresses. The authentication system requires the proper hostnames when authenticating users between the services.

Fill in the domain names or IP addresses of the components below and record the auto generated wakari password in the box below after installing the Wakari Server component.

Component	Name or IP address
Wakari Server	
Wakari Gateway	
Wakari Compute	

## Install Wakari Server

The Wakari server is the administrative front end to the system. This is where users login to the system, where user accounts are stored, and where admins can manage the system.

## Wakari Server Preparation Prerequisites

### Download Prerequisite RPMs

- **Air Gap Installation:** Copy RPMs from the Air Gap media
- **Regular Installation:**

```
RPM_CDN="https://820451f3d8380952ce65-4cc6343b423784e82fd202bb87cf87cf.ssl.cf1.rackcdn.com"
curl -O $RPM_CDN/nginx-1.6.2-1.el6ngx.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-tools-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-shell-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-server-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-mongos-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/mongodb-org-2.6.8-1.x86_64.rpm
curl -O $RPM_CDN/elasticsearch-1.7.2.noarch.rpm
curl -O $RPM_CDN/jre-8u65-linux-x64.rpm
```

### Install Prerequisite RPMs

```
sudo yum install -y *.rpm
sudo /etc/init.d/mongod start
sudo /etc/init.d/elasticsearch start
sudo chkconfig --add elasticsearch
```

## Run the Wakari Server Installer

### Set Variables and Change Permissions

```
export WAKARI_SERVER=<FQDN HOSTNAME> # Use the real FQDN
chmod a+x wakari-*.sh                # Set installer to be executable

sudo ./wakari-server-0.10.0-Linux-x86_64.sh -w $WAKARI_SERVER
```

### Run Wakari Server Installer

```
sudo ./wakari-server-0.10.0-Linux-x86_64.sh -w $WAKARI_SERVER
<license text>
...
...

PREFIX=/opt/wakari/wakari-server
Logging to /tmp/wakari_server.log
Checking server name
Ready for pre-install steps
Installing miniconda
...
...
Checking server name
Loading config from /opt/wakari/wakari-server/etc/wakari/config.json
Loading config from /opt/wakari/wakari-server/etc/wakari/wk-server-config.json

=====

Created password '<RANDOM_PASSWORD>' for user 'wakari'

=====

Starting Wakari daemons...
installation finished.
```

After successfully completing the installation script, the installer will create the administrator account (wakari user) and assign it a password:

```
Created password '<RANDOM_PASSWORD>' for user 'wakari'
```

**Record this password.** It will be needed in the following steps. It is also available in the installation log file found at `/tmp/wakari_server.log`

### Restart ElasticSearch

Restart elasticsearch to read the new config file

```
sudo service elasticsearch restart
```

### Test the Wakari Server install

Visit `http://$WAKARI_SERVER`. You should be shown the **"license expired"** page.

### Update the License

From the **"license expired"** page, follow the onscreen instructions to upload your license file. After submitting, you should see the login page.

## Install Wakari Gateway

The gateway is a reverse proxy that authenticates users and automatically directs them to the proper Wakari Compute machine for their project. Users will not notice this component as it automatically routes them.

## Set Variables and Change Permissions

```
export WAKARI_SERVER=<FQDN HOSTNAME> # Use the real FQDN
export WAKARI_GATEWAY_PORT=8080
export WAKARI_GATEWAY=<FQDN HOSTNAME> # will be needed shortly
chmod a+x wakari-*.sh # Set installer to be executable
```

## Run Wakari Gateway Installer

```
sudo ./wakari-gateway-0.10.0-Linux-x86_64.sh -w $WAKARI_SERVER
<license text>
...
...

PREFIX=/opt/wakari/wakari-gateway
Logging to /tmp/wakari_gateway.log
...
...
Checking server name
Please restart the Gateway after running the following command to connect this Gateway to the Wakari

PATH=/opt/wakari/wakari-gateway/bin:$PATH \
/opt/wakari/wakari-gateway/bin/wk-gateway-configure \
--server http://1.1.1.1 --host 1.1.1.2 --port 8080 --name Gateway \
--protocol http --summary Gateway --username wakari --password password
```

**NOTE:** replace **password** with the password of the wakari user that was generated during server installation.

## Start the Wakari Gateway

```
sudo service wakari-gateway start
```

## Register the Wakari Gateway

The Wakari Gateway needs to register with the Wakari Server. This needs to be authenticated, so the wakari user's credentials created during the Wakari Server install need to be used. **This needs to be run as root** to write the configuration file: `/opt/wakari/wakari-gateway/etc/wakari/wk-gateway-config.json`

```
PATH=/opt/wakari/wakari-gateway/bin:$PATH \
/opt/wakari/wakari-gateway/bin/wk-gateway-configure \
--server http://$WAKARI_SERVER --host $WAKARI_GATEWAY \
--port $WAKARI_GATEWAY_PORT --name Gateway --protocol http \
--summary Gateway --username wakari \
--password '<USE PASSWORD SET ABOVE>'
```

## Ensure Proper Permissions

```
sudo chown wakari /opt/wakari/wakari-gateway/etc/wakari/wk-gateway-config.json
```

## Restart the gateway to load the new configuration file

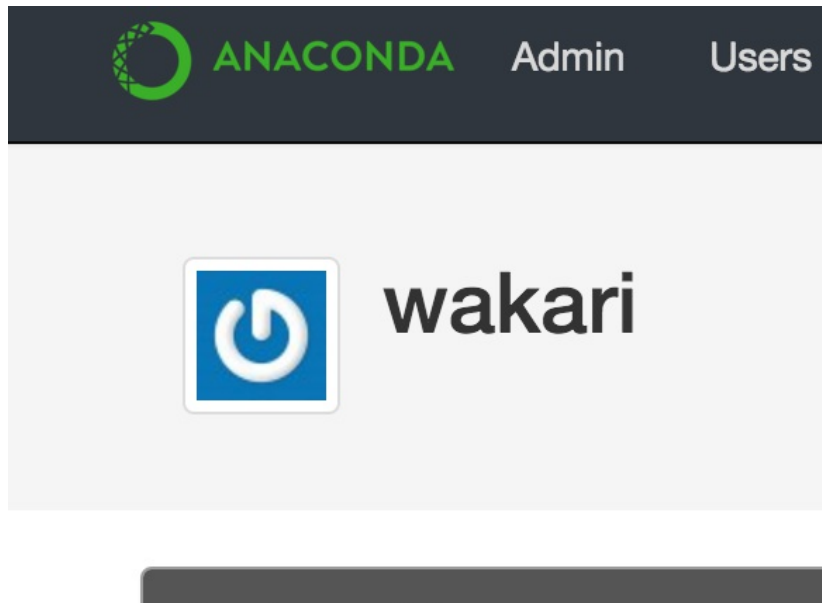
```
sudo service wakari-gateway restart
```

**NOTE:** Ignore any errors about missing `/lib/lsb/init-functions`

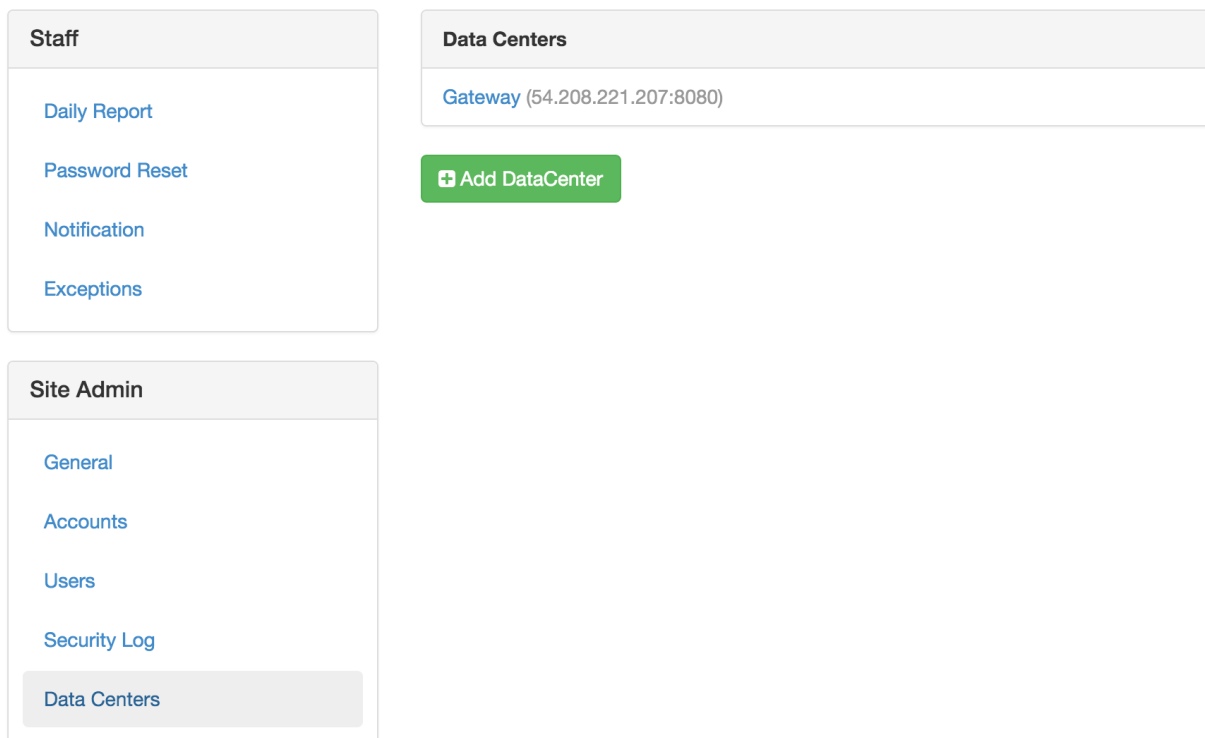
## Verify the Wakari Gateway has Registered

1. Login to the Wakari Server using Chrome or Firefox browser using the wakari user.

2. Click the Admin link in the toolbar



3. Click the Datacenters sub section and then click your datacenter:



4. Verify that your datacenter is registered and status is `{"status": "ok", "messages": []}`

**Datacenter Gateway**

Edit

**Provider**  
wk\_server:plugins.providers.enterprise

**Client ID**  
5705cd4233d6fd31f4e97fb4

**Client Secret**  
235fb27a-e6a2-48bc-a80f-30336ad912d5

**Redirect URIs**  
http://54.208.221.207:8080/login/authorized

**config.json**

```
{
  "CDN": "http://54.208.221.207/static/",
  "SUBDOMAIN_ROUTING": false,
  "client_id": "5705cd4233d6fd31f4e97fb4",
  "client_secret": "235fb27a-e6a2-48bc-a80f-30336ad912d5",
  "WAKARI_SERVER": "http://54.208.221.207",
  "port": 8080
}
```

**status**

```
{"status": "ok", "messages": []}
```

Back

Remove

## Install Wakari Compute

This is where projects are stored and run. Adding multiple Wakari Compute machines allows one to scale-out horizontally to increase capacity. Projects can be created on individual compute nodes to spread the load.

### Set Variables and Change Permissions

```
export WAKARI_SERVER=<FQDN HOSTNAME> # Use the real FQDN
chmod a+x wakari-*.sh                # Set installer to be executable
```

### Run Wakari Compute Installer

```
sudo ./wakari-compute-0.10.0-Linux-x86_64.sh -w $WAKARI_SERVER
...
...
PREFIX=/opt/wakari/wakari-compute
Logging to /tmp/wakari_compute.log
Checking server name
...
...
Initial clone of root environment...
Starting Wakari daemons...
installation finished.
Do you wish the installer to prepend the wakari-compute install location
to PATH in your /root/.bashrc ? [yes|no]
[no] >>> yes
```

### Configure Wakari Compute Node

Once installed, you need to configure the Compute Launcher on Wakari Server.

1. Point your browser at the Wakari Server
2. Login as the wakari user
3. Click on the Admin link in the top navbar
4. Click on Enterprise Resources in the left navbar
5. Click on Add Resource
6. Select the correct (probably the only) Data Center to associate this Compute Node with

7. For URL, enter **http://\$WAKARI\_COMPUTE:5002**.

**NOTE:** If the Compute Launcher is located on the same box as the Gateway, we recommend using **http://localhost:5002** for the URL value.

8. Add a Name and Description for the compute node
9. Click the Add Resource button to save the changes.

**Congratulations!** You've now successfully installed and configured Anaconda Enterprise Notebook.