

Deploy Ariclaw Website to Apache on stage-web-eg3

Overview

Provisioned a working Apache web server on stage-web-eg3, opened the required firewall ports, pulled the Ariclaw site from GitLab, and deployed the web content into /var/www/html with correct ownership. Verified successful delivery from server to browser.

Evidence (last 11 screenshots)

Enabled HTTP access in the firewall and reloaded rules.

Confirmed the web content/template/footer/source content on the server.

Opened ports 80/tcp and 443/tcp, confirmed services http/https, and validated active firewall configuration (including existing SSH rich rule).

Installed git on stage-web-eg3.

Cloned the ariclaw repository from GitLab.

Attempted deployment with rsync and identified rsync was missing, then installed it.

Successfully synced the site files into /var/www/html/ (incremental file list).

Set correct web directory ownership with chown -R apache:apache /var/www/html.

Restarted Apache and validated content locally with curl http://localhost (HTML returned).

Listed /var/www/html and confirmed expected site files/directories and apache:apache ownership.

Verified the Ariclaw homepage loads in a browser via the server IP, confirming end-to-end deployment.

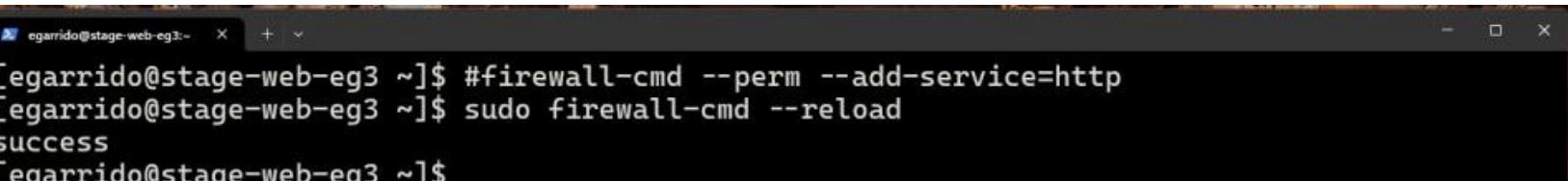
Result

Apache is running and serving the Ariclaw site correctly from /var/www/html over HTTP, firewall rules allow web traffic, and the deployment pipeline (GitLab → server clone → rsync to web root → permissions → service restart → browser validation) is confirmed working.

This screenshot confirms that the Apache HTTP Server (httpd) is already installed, enabled, and actively running on the stage-web-eg3 system. The dnf output verifies no additional installation actions were required, indicating the package was previously deployed. The systemctl status httpd output shows the service is enabled at boot, actively running, and managed by systemd, with multiple worker processes successfully spawned. Service logs confirm Apache started cleanly and is listening as expected, validating a stable and operational web service state.

```
[egarrido@stage-web-eg3 ~]$ sudo dnf install httpd -y
[sudo] password for egarrido:
Last metadata expiration check: 1:37:44 ago on Sat 27 Sep 2025 10:14:35 AM EDT.
Package httpd-2.4.62-7.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[egarrido@stage-web-eg3 ~]$ sudo systemctl enable --now httpd
[egarrido@stage-web-eg3 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Thu 2025-09-25 23:51:01 EDT; 1 day 12h ago
     Docs: man:httpd.service(8)
 Main PID: 19603 (httpd)
   Status: "Total requests: 1; Idle/Busy workers 100/0;Requests/sec: 7.71e-06; >
    Tasks: 177 (limit: 4604)
   Memory: 16.2M
      CPU: 2min 10.807s
     CGroup: /system.slice/httpd.service
             └─19603 /usr/sbin/httpd -DFOREGROUND
Sep 25 23:51:00 stage-web-eg3.procore.prod1 systemd[1]: Starting The Apache HTTP >
Sep 25 23:51:01 stage-web-eg3.procore.prod1 systemd[1]: Started The Apache HTTP S>
Sep 25 23:51:01 stage-web-eg3.procore.prod1 httpd[19603]: Server configured, list>
Lines 1-19/19 (END)
```

This screenshot shows the firewalld configuration reload on the stage-web-eg3 system. The HTTP service addition command is commented out, indicating no changes were made during this step. The subsequent firewall-cmd --reload command completes successfully, confirming that the firewall configuration was reloaded without errors and that the current firewall rules remain active and consistent.



A terminal window titled 'egarrido@stage-web-eg3 ~' showing the following command history:

```
[egarrido@stage-web-eg3 ~]$ #firewall-cmd --perm --add-service=http  
[egarrido@stage-web-eg3 ~]$ sudo firewall-cmd --reload  
success  
[egarrido@stage-web-eg3 ~]$
```

This screenshot documents the installation of Git on the stage-web-eg3.procore.prod1 server. After establishing an SSH session from the dev-app-eg3 host, the dnf install git -y command is executed. The output shows successful dependency resolution and installation of Git along with required supporting packages, confirming that version control tooling is now available on the system for source code management and repository operations.

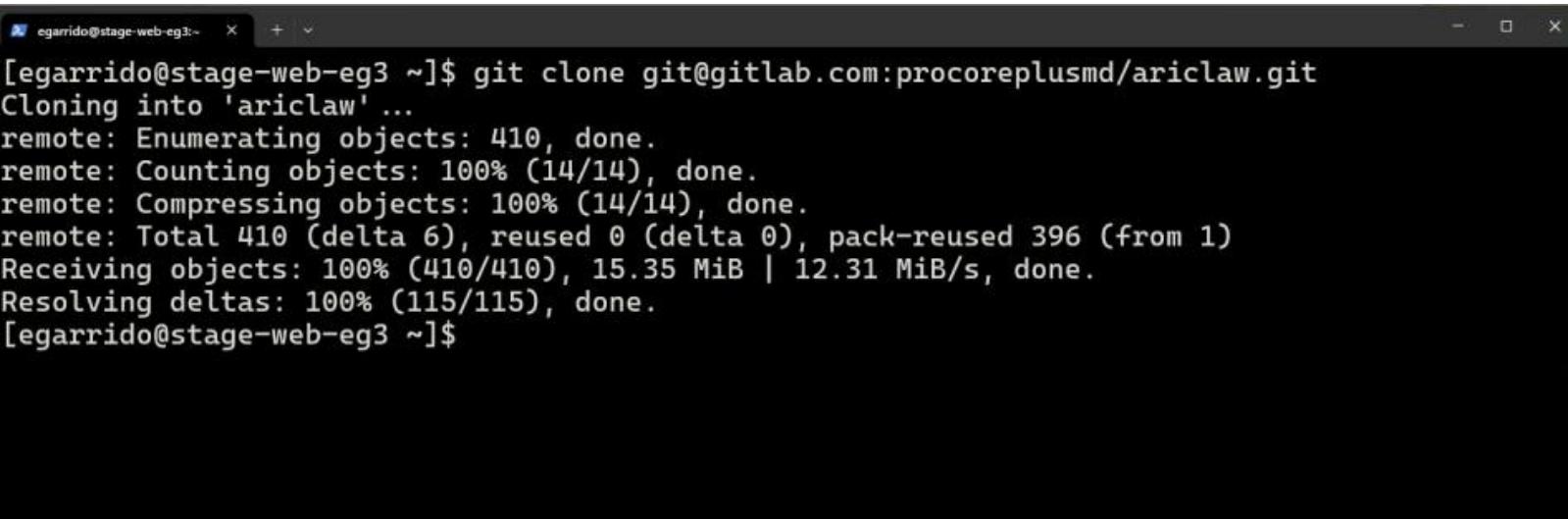
```
[egarrido@dev-app-eg3 ~]$ ssh egarrido@stage-web-eg3.procore.prod1
Last login: Sat Sep 27 12:22:14 2025 from 10.1.31.135
[egarrido@stage-web-eg3 ~]$ sudo dnf install git -y
[sudo] password for egarrido:
Last metadata expiration check: 2:21:15 ago on Sat 27 Sep 2025 10:14:35 AM EDT.
Dependencies resolved.



| Package                  | Architecture | Version       | Repository | Size  |
|--------------------------|--------------|---------------|------------|-------|
| Installing:              |              |               |            |       |
| git                      | x86_64       | 2.47.3-1.el9  | appstream  | 51 k  |
| Installing dependencies: |              |               |            |       |
| emacs-filesystem         | noarch       | 1:27.2-18.el9 | appstream  | 9.2 k |
| git-core                 | x86_64       | 2.47.3-1.el9  | appstream  | 4.7 M |
| git-core-doc             | noarch       | 2.47.3-1.el9  | appstream  | 3.0 M |
| perl-AutoLoader          | noarch       | 5.74-483.el9  | appstream  | 21 k  |
| perl-B                   | x86_64       | 1.80-483.el9  | appstream  | 180 k |
| perl-Carp                | noarch       | 1.50-460.el9  | appstream  | 30 k  |
| perl-Class-Struct        | noarch       | 0.66-483.el9  | appstream  | 22 k  |
| perl-Data-Dumper         | x86_64       | 2.174-462.el9 | appstream  | 56 k  |
| perl-Digest              | noarch       | 1.19-4.el9    | appstream  | 26 k  |
| perl-Digest-MD5          | x86_64       | 2.58-4.el9    | appstream  | 37 k  |
| perl-DynaLoader          | x86_64       | 1.47-483.el9  | appstream  | 25 k  |


```

This screenshot shows a successful Git repository clone operation on the stage-web-eg3 server. The git clone command is used to pull the ariclaw repository from a GitLab remote source. The output confirms that all objects were enumerated, downloaded, and unpacked correctly, with deltas resolved and no errors reported. This verifies network connectivity to GitLab and confirms that the server can retrieve and work with external source code repositories.



A screenshot of a terminal window titled "egarrido@stage-web-eg3 ~". The window displays the output of a "git clone" command. The text output is as follows:

```
[egarrido@stage-web-eg3 ~]$ git clone git@gitlab.com:procoreplusmd/ariclaw.git
Cloning into 'ariclaw' ...
remote: Enumerating objects: 410, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 410 (delta 6), reused 0 (delta 0), pack-reused 396 (from 1)
Receiving objects: 100% (410/410), 15.35 MiB | 12.31 MiB/s, done.
Resolving deltas: 100% (115/115), done.
[egarrido@stage-web-eg3 ~]$
```

This screenshot documents the deployment preparation step on the stage-web-eg3 server where rsync is required to publish application files to the Apache web root.

An initial attempt to run rsync fails because the utility is not installed.

The missing dependency is identified immediately.

rsync is installed successfully using dnf, confirming proper repository access and package management on CentOS Stream 9.

The system is now ready to synchronize application content from the cloned Git repository into /var/www/html/ for web serving.

```
[egarrido@stage-web-eg3 ~]$ sudo rsync -av ariclaw/ /var/www/html/
sudo: rsync: command not found
[egarrido@stage-web-eg3 ~]$ sudo dnf install rsync -y
Last metadata expiration check: 2:27:37 ago on Sat 27 Sep 2025 10:14:35 AM EDT.
Dependencies resolved.

Transaction Summary

Install 1 Package

Total download size: 408 k
Installed size: 767 k
Downloading Packages:
rsync-3.2.5-3.el9.x86_64.rpm          329 kB/s | 408 kB   00:01

Total                                         279 kB/s | 408 kB   00:01

Running transaction check
Transaction check succeeded
```

An attempt is made to copy application files into /var/www/html/ using rsync, but the command initially fails because the rsync utility is not installed on the system. The session then documents the installation of rsync via dnf, showing successful package resolution, download, and installation, preparing the server to perform file synchronization for the web application.

```
[egarrido@stage-web-eg3 ~]$ sudo rsync -av ariclaw/ /var/www/html/
sudo: rsync: command not found
[egarrido@stage-web-eg3 ~]$ sudo dnf install rsync -y
Last metadata expiration check: 2:27:37 ago on Sat 27 Sep 2025 10:14:35 AM EDT.
Dependencies resolved.



| Package     | Architecture | Version     | Repository | Size  |
|-------------|--------------|-------------|------------|-------|
| Installing: |              |             |            |       |
| rsync       | x86_64       | 3.2.5-3.el9 | baseos     | 408 k |


Transaction Summary
Install 1 Package

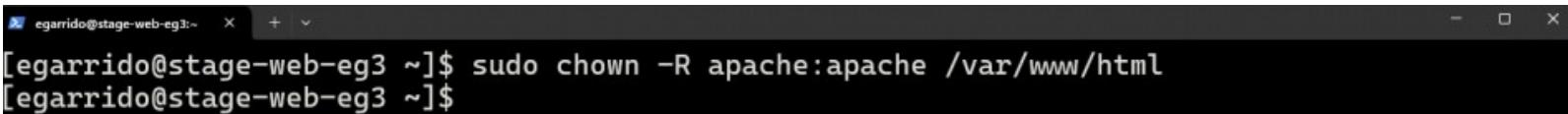
Total download size: 408 k
Installed size: 767 k
Downloading Packages:
rsync-3.2.5-3.el9.x86_64.rpm          329 kB/s | 408 kB   00:01

Total                                         279 kB/s | 408 kB   00:01
Running transaction check
Transaction check succeeded
```

The command successfully runs rsync -av ariclaw/ /var/www/html/, transferring website content into the Apache web root. The output lists multiple HTML files and a PHP file being copied, along with the .git directory and its internal metadata, confirming that the application files and repository contents were synchronized to the server without errors.

```
egarrido@stage-web-eg3:~$ sudo rsync -av ariclaw/ /var/www/html/
sending incremental file list
./
about.html
attorneys.html
blog.html
contact.html
contact_process.php
elements.html
index.html
service.html
single-blog.html
team.html
.git/
.git/HEAD
.git/config
.git/description
.git/index
.git/packed-refs
.git/branches/
.git/hooks/
```

The command recursively updates ownership of the /var/www/html directory, assigning both the user and group to apache. This ensures the Apache web server has proper permissions to read and manage the deployed web content, completing the post-deployment permission configuration step.



```
[egarrido@stage-web-eg3 ~]$ sudo chown -R apache:apache /var/www/html
[egarrido@stage-web-eg3 ~]$
```

The Apache HTTP service is restarted, and a local curl request to `http://localhost` successfully returns the HTML source of the deployed website. The output confirms that the web server is serving the expected application content, including the document structure, metadata, page title, and linked CSS assets, validating that the site is accessible and functioning correctly through Apache.

```
[egarrido@stage-web-eg3 ~]$ sudo systemctl restart httpd
[egarrido@stage-web-eg3 ~]$ curl http://localhost
<!doctype html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>ariclaw</title>
    <link rel="icon" href="img/favicon.png">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <!-- animate CSS -->
    <link rel="stylesheet" href="css/animate.css">
    <!-- owl carousel CSS -->
    <link rel="stylesheet" href="css/owl.carousel.min.css">
    <!-- themify CSS -->
    <link rel="stylesheet" href="css/themify-icons.css">
    <!-- flaticon CSS -->
    <link rel="stylesheet" href="css/flaticon.css">
    <!-- font awesome CSS -->
    <link rel="stylesheet" href="css/magnific-popup.css">
```

A detailed directory listing of /var/www/html confirms that the web application files have been successfully deployed and are owned by the apache user and group. The output shows standard web permissions applied consistently across HTML pages, PHP scripts, and supporting asset directories such as css, js, img, fonts, and webfonts. Core site files including index.html, about.html, services.html, and related content are present with executable read permissions, indicating the Apache service can properly access and serve the website content.

```
[egarrido@stage-web-eg3 ~]$ ls -l /var/www/html
total 276
-rwxr-xr-x. 1 apache apache 21719 Sep 27 12:37 about.html
drwxr-xr-x. 7 apache apache 95 Sep 27 12:37 'Ariclaw Lawyer -DOC'
-rwxr-xr-x. 1 apache apache 24824 Sep 27 12:37 attorneys.html
-rwxr-xr-x. 1 apache apache 29219 Sep 27 12:37 blog.html
-rwxr-xr-x. 1 apache apache 13730 Sep 27 12:37 contact.html
-rwxr-xr-x. 1 apache apache 1416 Sep 27 12:37 contact_process.php
drwxr-xr-x. 2 apache apache 4096 Sep 27 12:37 css
-rwxr-xr-x. 1 apache apache 36290 Sep 27 12:37 elements.html
drwxr-xr-x. 2 apache apache 166 Sep 27 12:37 fonts
drwxr-xr-x. 12 apache apache 4096 Sep 27 12:37 img
-rwxr-xr-x. 1 apache apache 33847 Sep 27 12:37 index.html
drwxr-xr-x. 2 apache apache 4096 Sep 27 12:37 js
drwxr-xr-x. 2 apache apache 4096 Sep 27 12:37 sass
-rwxr-xr-x. 1 apache apache 20759 Sep 27 12:37 service.html
-rwxr-xr-x. 1 apache apache 31888 Sep 27 12:37 single-blog.html
-rwxr-xr-x. 1 apache apache 23162 Sep 27 12:37 team.html
drwxr-xr-x. 2 apache apache 4096 Sep 27 12:37 webfonts
[egarrido@stage-web-eg3 ~]$
```

The Ariclaw website loads successfully over HTTP, displaying the homepage with full styling and navigation intact. The landing section presents the firm's branding, headline message, call-to-action button, and hero image, confirming that Apache is serving the deployed web content correctly and that static assets (CSS, images, and fonts) are accessible. This validates end-to-end web deployment from the server to the browser

A screenshot of a web browser showing the Ariclaw website. The address bar indicates the site is not secure (10.1.31.136). The page features a large hero image of a man in a suit. To the left of the image, there is a headline and a call-to-action button. The navigation menu includes Home, About, Service, Attorneys, Blog, Contact, and Let's Talk. A small gavel icon is visible near the top left of the hero image.

ariclaw

Not secure 10.1.31.136

Home About Service Attorneys Blog Contact Let's Talk

Ariclaw

Finest And Strongest Law Firm Win The World

Subdue which man creeping was image you dry lesser every live our be gree male may living beginning appear moveth beast

More About Us

The lawyers truth is not truth but consistency or a consistent expediency

Summary

The screen recordings document the full deployment and validation of the Ariclaw website on the stage-web-eg3 Apache server. The process begins by confirming firewall rules allow HTTP and HTTPS traffic, followed by preparing the system with required utilities such as git and rsync. The Ariclaw source code is then cloned from GitLab and synchronized into /var/www/html, placing all website assets in the correct web root.

File ownership and permissions are adjusted to match the Apache service account, ensuring the web server can properly read and serve the content. The httpd service is restarted and verified as active and running. Server-side testing confirms successful content delivery by retrieving HTML output directly from localhost.

The workflow concludes with validation of the deployed files through a directory listing and a browser-based test using the server's IP address, confirming the Ariclaw homepage loads correctly and the web application is fully operational in the staging environment.

Sanitation: Sensitive information such as internal IP addresses, hostnames, usernames, and repository identifiers has been intentionally sanitized to prevent disclosure of environment-specific or private details.