CONFIGURATION GUIDES › DOCKER

Deploy the agent in the Nginx-proxy container

Applicare 9.2 · Updated February 27, 2025 · docker, nginx · **Pending technical review**

Note: This step is only necessary when initially creating the container that deploys the agent. For subsequent operations, such as stopping and restarting the container, the steps below are not needed.

Step 1: Edit the default.conf file and comment out the existing log format and replace it with the following log_format.

```
TEXT
log_format main escape=default '$remote_addr – $remote_user $time_iso8601 [$request_time]
"$request" $status $body_bytes_sent "$http_referer" "$http_user_agent"
"$http_x_forwarded_for" "$upstream_addr" $host';
```

```
gzip_types text/plain text/css application/javascript application/json application/x-javascript text/xml application/xml application/xml+rss
#log_format vhost escape=default '$host $remote_addr - $remote_user [$time_local] "$request" $status $body_bytes_sent "$http_referer" "$http
log_format vhost escape=default '$remote_addr - $remote_user $time_iso8601 [$request_time] "$request" $status $body_bytes_sent "$http_refere
access_log off;
```

Step 2: If there is a line called **access_log off**, comment out that line and add it to the below condition in the next line.

```
TEXT
access_log /var/log/nginx/access.log main;
access_log /dev/stdout main;
```

```
    log_format vhost escape=default '$remote_addr - $remote_user $time_iso8601 [$request_time] "$request"
    #access_log off;
    access_log /var/log/nginx/access.log vhost;
    access_log /dev/stdout vhost;
        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA
        ssl_prefer_server_ciphers off;
    error_log /dev/stderr;
```

Note: If the sub_status module is available, proceed with Step 3 as outlined. However, if the module is not available, skip Step 3 and proceed directly to Step 4.

Step 3: Add the following lines into the http server module.

```
YAML
Syntax:
location /nginx-status {stub_status on;allow IPADDRESS;
deny all;}

Example:
location /nginx-status {stub_status on;allow 192.168.1.7;
allow  all;}
```

```
server {
    server_name _; # This is just an invalid value which will never trigger on a real hostname.
    server_tokens off;
    access_log /var/log/nginx/access.log vhost;
    access_log /dev/stdout vhost;
    http2 on;
    listen 80;
    listen 443 ssl;
    ssl_session_cache shared:SSL:50m;
    ssl_session_tickets off;
    # No default certificate found, so reject SSL handshake;
    ssl_reject_handshake on;
    location ^~ / {
        return 503;
    }
    location /nginx-status {
        stub_status on;
        allow 192.168.1.8;
        allow all;
        }
}
```

Step 4: Download the file **agent_deploy.sh** ***(sample file, available on the documentation site)*** from this link. Edit the file and change the **protocol**, **IP**, and **port** as shown below.

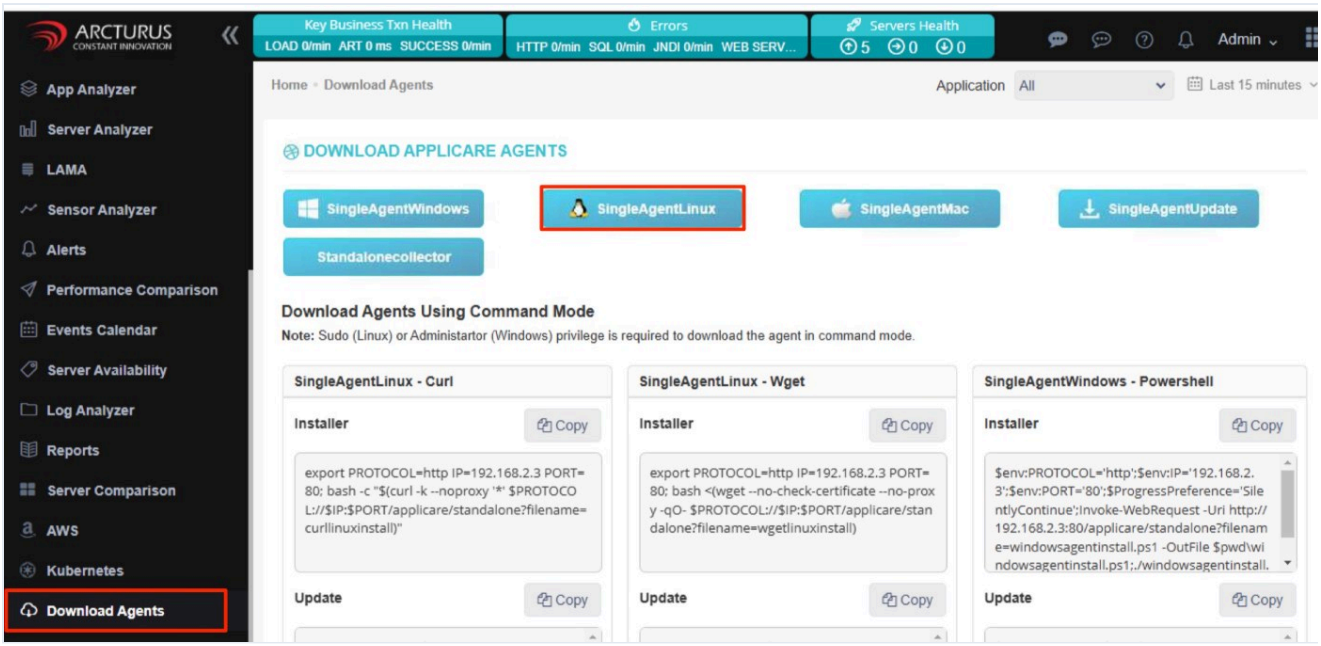
```
TEXT
PROTOCOL="<Applicare_protocol>"
IP="<Applicare_IP>"
PORT="<Applicare_port>"
```

<Applicare_protocol>= The protocol used by the Applicare controller is http or https.

<Applicare_IP>**= IP of the machine running Applicare controller**

<Applicare_port>= Applicare controller running port

Step 5: Access the **Applicare console**. Go to the **Download Agent** screen in the left-side menu, Download the **SingleAgentLinux** agent and extract it into the designated directory. **ApplicareSingleAgentLinux.**



Step 6: Add the following command along with the command that starts the container.

Add the following command before the image name.

```
TEXT
-v where the agent_deploy.sh file is located/agent_deploy.sh:/usr/local/agent_deploy.sh
-v ApplicareSingleAgentLinuxDirectory:/usr/local/ApplicareSingleAgentLinux
--entrypoint sh
--hostname The name which we need to add the agent in Applicare
```

Add the following command after the image name.

```
TEXT
-c 'rm /var/log/nginx/access.log && touch /var/log/nginx/access.log && openssl dhparam -out
/etc/nginx/dhparam/dhparam.pem 2048  && chown nginx:nginx /var/log/nginx/access.log && chmod
+x /usr/local/agent_deploy.sh && /usr/local/agent_deploy.sh && nginx -g "daemon off;"'
```

Example:

```
BASH
docker run -d --name nginx -v /home/nginx/nginx.conf:/etc/nginx/nginx.conf:ro -v
/var/run/docker.sock:/tmp/docker.sock:ro  -p 81:80 -p 443:443 -v
/home/nginx/agent_deploy.sh:/usr/local/agent_deploy.sh -v
/root/ApplicareSingleAgentLinux:/usr/local/ApplicareSingleAgentLinux  --entrypoint sh --
hostname jpetstore jwilder/nginx-proxy -c 'rm /var/log/nginx/access.log && touch
/var/log/nginx/access.log && openssl dhparam -out /etc/nginx/dhparam/dhparam.pem 2048 &&
chown nginx:nginx /var/log/nginx/access.log && chmod +x /usr/local/agent_deploy.sh &&
/usr/local/agent_deploy.sh && nginx -g "daemon off;"'
```

Note:

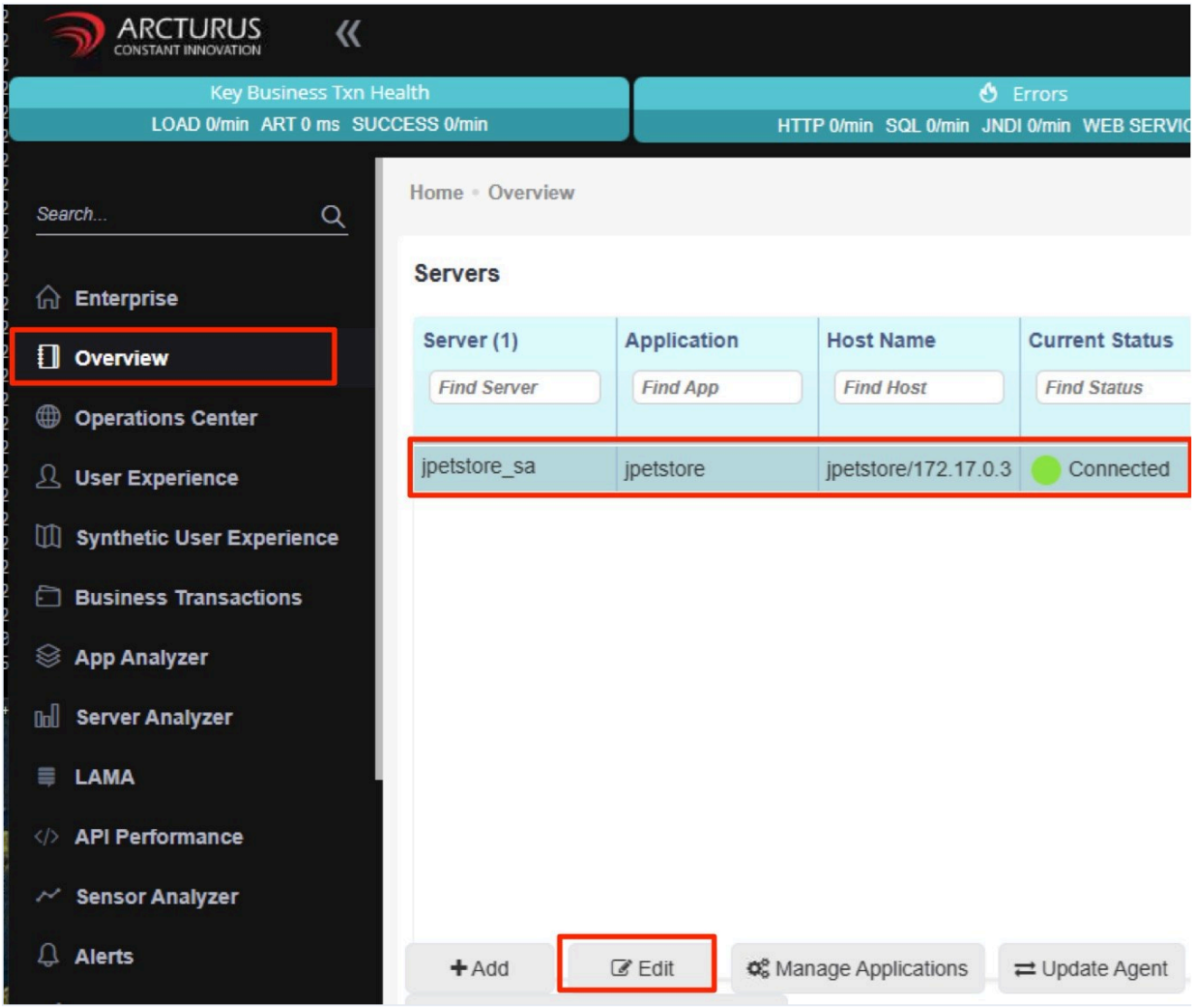
rm /var/log/nginx/access.log= Deletes the file or symbolic link at /var/log/nginx/access.log

touch /var/log/nginx/access.log= Creates a new, empty file at /var/log/nginx/access.log if it doesn't already exist.

openssl dhparam -out /etc/nginx/dhparam/dhparam.pem 2048= This command creates the DH parameters essential for SSL/TLS encryption, to enhancing secure communication and ensuring forward secrecy.

chown nginx:nginx /var/log/nginx/access.log= Changes the ownership of /var/log/nginx/access.log to the nginx user and group.

Step 7: Once the container starts, the agent will be automatically added to the Applicare controller. To manage it, log in to the Applicare console and go to the Overview menu. Find the newly added agent, select it, and click **Edit** to configure as needed.



Note: If the sub_status module is available, proceed with Step 8 as outlined. However, if the module is not available, skip Step 8 and proceed directly to Step 9.

Step 8: Enable the "**Add Nginx Configuration**" checkbox and add the below values.

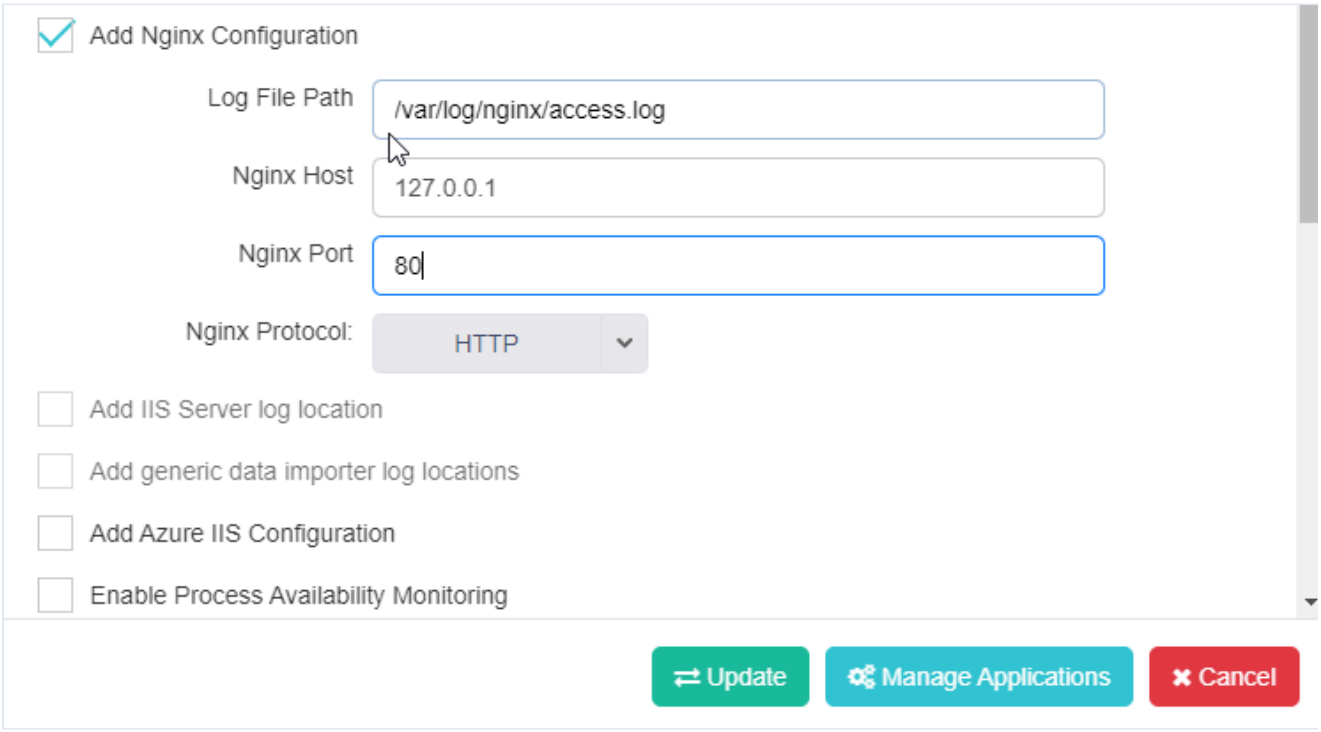
1. Log file path

2. Nginx Host

3. Nginx Port

4. Nginx Protocol

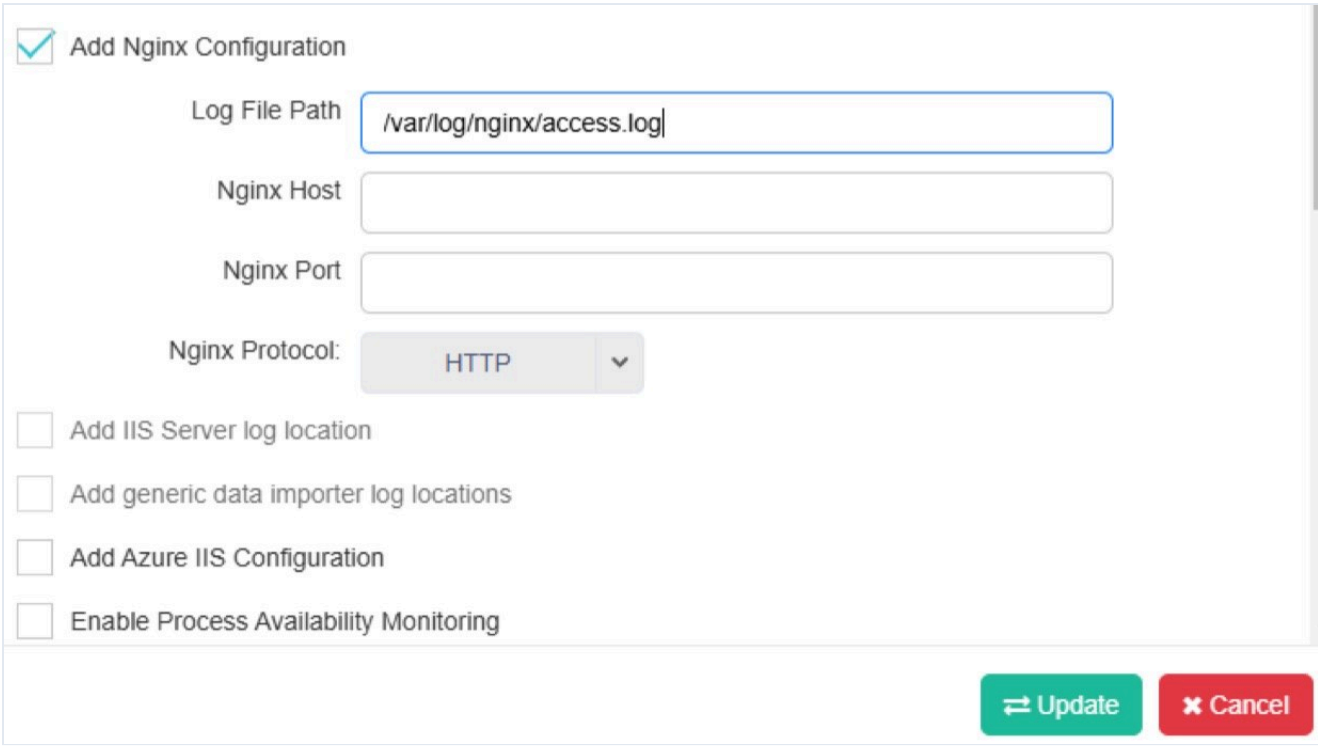
- Click the **update** button to save.



Step 9: Enable the "**Add Nginx Configuration**" checkbox and add the below values.

1. Log file path

- Click the **update** button to save.


Add Nginx Configuration
Log File Path
/var/log/nginx/access.log
Nginx Host
Nginx Port
Nginx Protocol:
HTTP
Add IIS Server log location
Add generic data importer log locations
Add Azure IIS Configuration
Enable Process Availability Monitoring
Update
Cancel