Proxmox container 'nextcloud.test'

Status

- läuft, beta

Container setup

- image: debian10 with users
- 1. Netzwerkinterface
  - eth0
  - 10.2.0.20/24 DG: 10.2.0.254; fd00:10:2:0::20/64 DGfd00:10:2:0::0
  - bridge: vmbr0
- 2. Netzwerkinterface:
  - eth1,
  - 10.3.0.20/24 DG: 10.3.0.254; fd00:10:3:0::20/64 DGfd00:10:3:0::0
  - bridge: vmbr1

Ansible setup

- verbindet man sich als unpriviligierter user und möchte ein Command als ein anderer, unpriviligierter user ausführen (z.B. www-data), benötigt man

  ```
  allow_world_readable_tmpfiles = yes
  ```

  in der ansible.cfg, damit dies nicht als Fehler zum Abbruch führt.

Ansible Script

```
#!/usr/bin/env ansible-playbook

- name: 'install nextcloud'
  hosts: nextcloud
  become: true

  tasks:
  - name: remove apache2
    apt:
      name: apache2
      state: absent

  - name: install packages
    apt:
      pkg:
        - php7.3
        - php7.3-curl
        - php7.3-gd
        - php7.3-json
        - php7.3-xml
        - php7.3-mbstring
        - php7.3-zip
        - php7.3-mysql
        - php7.3-bz2
        - php7.3-intl
        - php7.3-redis
        - php7.3-imagick
        - php7.3-fpm
        - ffmpeg
```

```
        - nginx

  - name: check if nextcloud is already installed
    stat:
      path: /var/www/nextcloud
    register: nextcloud_exists

  - name: Download nextcloud and unpack
    unarchive:
      src: https://download.nextcloud.com/server/releases/nextcloud-20.0.0.zip
      dest: /var/www
      owner: www-data
      group: www-data
      remote_src: yes
    when: not nextcloud_exists.stat.exists

  - name: Remove file packed file
    file:
      path: /var/www/nextcloud-20.0.0.zip
      state: absent

  - name: first setup nextcloud
    become_user: www-data
    become: yes
    shell: |
      php occ  maintenance:install --database mysql --database-host 10.3.0.100 --database-
name nextcloud --database-port 3306  --database-user nc_user --database-pass 'dbpass' --
admin-user admin --admin-pass 'ampass' --data-dir /var/www/nextcloud/data
    args:
      chdir: /var/www/nextcloud/
      creates: /var/www/nextcloud/config/config.php

  - name: add domain to trusted domains
    become_user: www-data
    become: yes
    lineinfile:
      path: /var/www/nextcloud/config/config.php
      insertafter: .*localhost.*
      line: "    1 => 'bytecluster0002.bytespeicher.org'"
      firstmatch: yes

  - name: copy nginx config
    copy:
      src: ./conf/nextcloud.conf
      dest: /etc/nginx/sites-available/
      owner: root
      group: root
      mode: '0744'

  - name: Create a symbolic link
    file:
      src: /etc/nginx/sites-available/nextcloud.conf
      dest: /etc/nginx/sites-enabled/nextcloud.conf
      owner: root
      group: root
      state: link

  - name: load php-fpm
    systemd:
      state: restarted
      name: php7.3-fpm
```

```
  - name: Restart nginx
    systemd:
      state: restarted
      daemon_reload: yes
      name: nginx
```

Notizen

- Ansible script läd momentan das .zip nicht herunter. Fehler unbekannt.
- Sonderzeichen in Passwörtern führen zu fehlern. Genaue Escape-Sequence noch unbekannt.

Nginx Config

```
upstream php-handler {
    #server 127.0.0.1:9000;
    server unix:/var/run/php/php7.3-fpm.sock;
}

server {
    listen 8087;
    listen [::]:8087;
    server_name cloud.technikkultur-erfurt.de;

    # Enforce HTTPS
    #return 301 https://$server_name$request_uri;
#}

#server {
#    listen 443      ssl http2;
#    listen [::]:443 ssl http2;
#    server_name cloud.technikkultur-erfurt.de;

    # Use Mozilla's guidelines for SSL/TLS settings
    # https://mozilla.github.io/server-side-tls/ssl-config-generator/
    # ssl_certificate     /etc/ssl/nginx/cloud.example.com.crt;
    # ssl_certificate_key /etc/ssl/nginx/cloud.example.com.key;

    # HSTS settings
    # WARNING: Only add the preload option once you read about
    # the consequences in https://hstspreload.org/. This option
    # will add the domain to a hardcoded list that is shipped
    # in all major browsers and getting removed from this list
    # could take several months.
    #add_header Strict-Transport-Security "max-age=15768000; includeSubDomains; preload;"
always;

    # set max upload size
    client_max_body_size 512M;
    fastcgi_buffers 64 4K;

    # Enable gzip but do not remove ETag headers
    gzip on;
    gzip_vary on;
    gzip_comp_level 4;
    gzip_min_length 256;
    gzip_proxied expired no-cache no-store private no_last_modified no_etag auth;
    gzip_types application/atom+xml application/javascript application/json
application/ld+json application/manifest+json application/rss+xml application/vnd.geo+json
application/vnd.ms-fontobject application/x-font-ttf application/x-web-app-manifest+json
```

```
application/xhtml+xml application/xml font/opentype image/bmp image/svg+xml image/x-icon
text/cache-manifest text/css text/plain text/vcard text/vnd.rim.location.xloc text/vtt
text/x-component text/x-cross-domain-policy;

    # Pagespeed is not supported by Nextcloud, so if your server is built
    # with the `ngx_pagespeed` module, uncomment this line to disable it.
    #pagespeed off;

    # HTTP response headers borrowed from Nextcloud `.htaccess`
    add_header Referrer-Policy                      "no-referrer"   always;
    add_header X-Content-Type-Options               "nosniff"       always;
    add_header X-Download-Options                   "noopen"        always;
    add_header X-Frame-Options                      "SAMEORIGIN"    always;
    add_header X-Permitted-Cross-Domain-Policies    "none"          always;
    add_header X-Robots-Tag                         "none"          always;
    add_header X-XSS-Protection                     "1; mode=block" always;

    # Remove X-Powered-By, which is an information leak
    fastcgi_hide_header X-Powered-By;

    # Path to the root of your installation
    root /var/www/nextcloud;

    # Specify how to handle directories -- specifying `/index.php$request_uri`
    # here as the fallback means that Nginx always exhibits the desired behaviour
    # when a client requests a path that corresponds to a directory that exists
    # on the server. In particular, if that directory contains an index.php file,
    # that file is correctly served; if it doesn't, then the request is passed to
    # the front-end controller. This consistent behaviour means that we don't need
    # to specify custom rules for certain paths (e.g. images and other assets,
    # `/updater`, `/ocm-provider`, `/ocs-provider`), and thus
    # `try_files $uri $uri/ /index.php$request_uri`
    # always provides the desired behaviour.
    index index.php index.html /index.php$request_uri;

    # Default Cache-Control policy
    expires 1m;

    # Rule borrowed from `.htaccess` to handle Microsoft DAV clients
    location = / {
        if ( $http_user_agent ~ ^DavClnt ) {
            return 302 /remote.php/webdav/$is_args$args;
        }
    }

    location = /robots.txt {
        allow all;
        log_not_found off;
        access_log off;
    }

    # Make a regex exception for `/.well-known` so that clients can still
    # access it despite the existence of the regex rule
    # `location ~ /(\.|autotest|...)` which would otherwise handle requests
    # for `/.well-known`.
    location ^~ /.well-known {
        # The following 6 rules are borrowed from `.htaccess`

        rewrite ^/\.well-known/host-meta\.json  /public.php?service=host-meta-json  last;
        rewrite ^/\.well-known/host-meta        /public.php?service=host-meta       last;
        rewrite ^/\.well-known/webfinger        /public.php?service=webfinger       last;
```

```
        rewrite ^/\.well-known/nodeinfo         /public.php?service=nodeinfo        last;

        location = /.well-known/carddav     { return 301 /remote.php/dav/; }
        location = /.well-known/caldav      { return 301 /remote.php/dav/; }

        try_files $uri $uri/ =404;
    }

    # Rules borrowed from `.htaccess` to hide certain paths from clients
    location ~ ^/(?:build|tests|config|lib|3rdparty|templates|data)(?:$|/)  { return 404; }
    location ~ ^/(?:\.|autotest|occ|issue|indie|db_|console)              { return 404; }

    # Ensure this block, which passes PHP files to the PHP process, is above the blocks
    # which handle static assets (as seen below). If this block is not declared first,
    # then Nginx will encounter an infinite rewriting loop when it prepends `/index.php`
    # to the URI, resulting in a HTTP 500 error response.
    location ~ \.php(?:$|/) {
        fastcgi_split_path_info ^(.+?\.php)(/.*)$;
        set $path_info $fastcgi_path_info;

        try_files $fastcgi_script_name =404;

        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $path_info;
        fastcgi_param HTTPS on;

        fastcgi_param modHeadersAvailable true;         # Avoid sending the security headers
twice
        fastcgi_param front_controller_active true;     # Enable pretty urls
        fastcgi_pass php-handler;

        fastcgi_intercept_errors on;
        fastcgi_request_buffering off;
    }

    location ~ \.(?:css|js|svg|gif)$ {
        try_files $uri /index.php$request_uri;
        expires 6M;         # Cache-Control policy borrowed from `.htaccess`
        access_log off;     # Optional: Don't log access to assets
    }

    location ~ \.woff2?$ {
        try_files $uri /index.php$request_uri;
        expires 7d;         # Cache-Control policy borrowed from `.htaccess`
        access_log off;     # Optional: Don't log access to assets
    }

    location / {
        try_files $uri $uri/ /index.php$request_uri;
    }
}
```