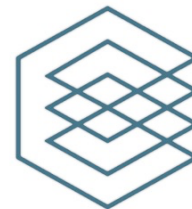


Vanity URLs

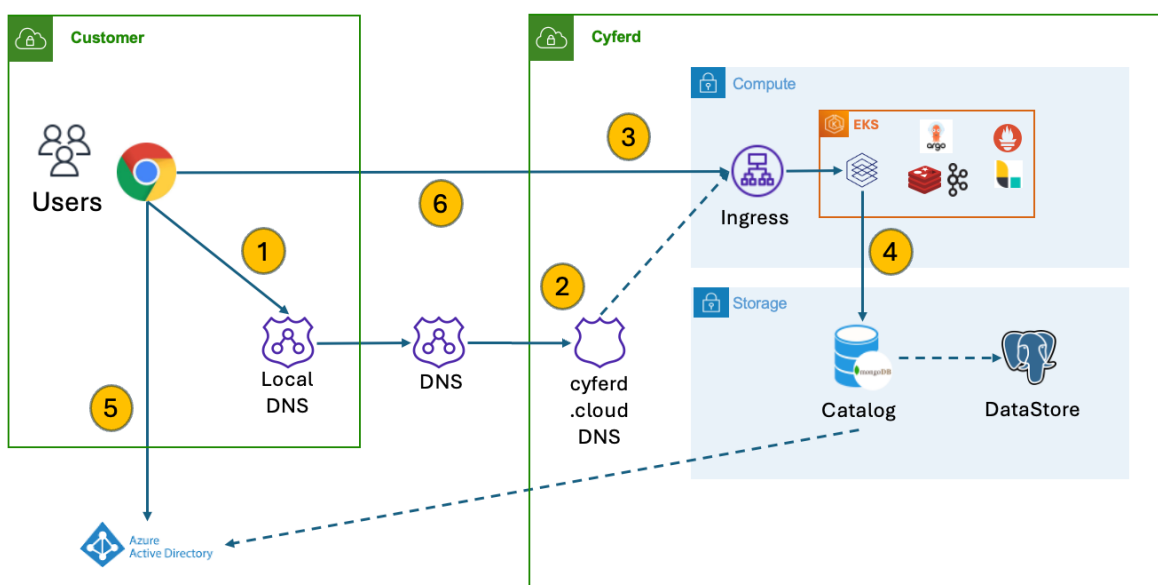


By “Vanity URLs” we mean the use of a non-standard URL to access a Cyferd tenancy.

All tenancies are identified in the Catalog by their HTTP Origin; normally this is **tenant.cyferd.cloud**. Some customers want to be able to rebrand their URL so that users browse to e.g. **cyferd.tenant.com**, or instead of accessing mod-uk.cyferd.cloud, they’d access cyferd.mod.uk

This is feasible, however the customer/tenant will be in control of the SSL Certificate that encrypts traffic to sites in their domain, so they will have to provide some infrastructure themselves, and adjust their OIDC Authentication Provider to recognize the alternate URL.

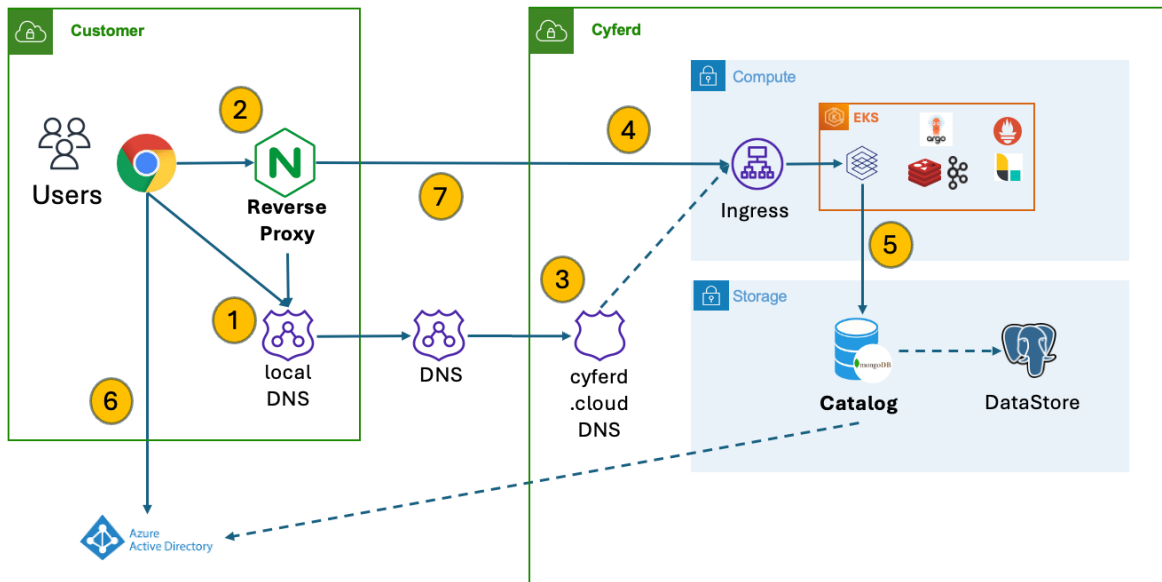
The normal flow of connection to Cyferd is



1. Browser asks local DNS for address of **tenant.cyferd.cloud**
2. DNS forwards request to authoritative zone, which resolves tenant.cyferd.cloud to an address of Cyferd’s Ingress Load Balancer
3. Browser makes TLS connection to Cyferd Load Balancer, passing HTTP Origin **tenant.cyferd.cloud**
4. Cyferd looks up **tenant.cyferd.cloud** in the Catalog to locate the Tenant configuration, which identifies the OIDC Authentication provider that the browser is redirected to
5. User authenticates with their OIDC Provider, which has whitelisted the **https://tenant.cyferd.cloud/api/callback** Redirect URI
6. Browser is redirected to <https://tenant.cyferd.cloud/>

To support a Vanity URL, the customer must provide a Reverse Proxy that is configured to forward connections to Cyferd. They must also ensure that their OIDC Authentication Provider has whitelisted the URL used to access the Reverse Proxy.

Cyferd must be advised of the Vanity URL so that this can be recorded in the Catalog.



1. Browser asks local DNS for address of **cyferd.tenant.com** which resolves cyferd.tenant.com to the address of a local Reverse Proxy (eg Nginx, Apache2)
2. Browser makes TLS connection to Reverse Proxy, passing HTTP Origin cyferd.tenant.com
3. Reverse Proxy is configured to forward this connection to ingress-xx.cyferd.cloud, and DNS resolves this to an address of Cyferd’s Ingress Load Balancer
4. Reverse Proxy connects to Cyferd Load Balancer, passing HTTP Origin **cyferd.tenant.com**
5. Cyferd looks up **cyferd.tenant.com** in Catalog to locate the Tenant configuration, which identifies the OIDC Authentication provider that the browser is redirected to
6. User authenticates with their OIDC Provider, which has whitelisted the **https://cyferd.tenant.com/api/callback** Redirect URI
7. Browser is redirected to <https://cyferd.tenant.com/>

The Reverse Proxy must be configured to upgrade /api/websocket connections to use a websocket, and all other traffic is simply forwarded to Cyferd.

Example configuration `/etc/nginx/sites-available/cyferd` for Nginx:

```
# so that the Connection header is correctly set to close when the
Upgrade header in the request is set to ''.
map $http_upgrade $connection_upgrade {
    default upgrade;
    '' close;
}

server {
    listen 443 ssl;

    # Tenant-supplied SSL Certificate
    ssl_certificate      /etc/nginx/ssl/SSL_certificate_tenant_com.crt;
    ssl_certificate_key  /etc/nginx/ssl/SSL_certificate_tenant_com.key;

    # Self Signed Certificates not supported for Websocket connections
    #ssl_certificate      /etc/ssl/certs/ssl-cert-snakeoil.pem;
    #ssl_certificate_key  /etc/ssl/private/ssl-cert-snakeoil.key;

    server_name cyferd.tenant.com;

    access_log  /var/log/nginx/access_cyferd.log;
    error_log   /var/log/nginx/error_cyferd.log;

    root /var/www/html;

    proxy_redirect off;
    proxy_http_version 1.1;

    # websocket (see http://nginx.org/en/docs/http/websocket.html)
    location /api/websocket {
        proxy_pass https://ingress-us.cyferd.cloud/api/websocket;
        proxy_http_version 1.1;
        proxy_set_header X-Forwarded-Host $http_host;
        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
    }

    location / {
        proxy_pass https://ingress-us.cyferd.cloud/;
        proxy_http_version 1.1;
        proxy_set_header X-Forwarded-Host $http_host;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```