Configuring a Redirect to OAC OM and OAC OM on OCI URL using Vanity URL

OAC – Classic (Customer Managed) with and without LBaaS, we know how to setup a Vanity URL

OAC-Classic : What are the Steps to setup custom SSL certificates for Oracle Analytics Cloud-Classic (OAC-Classic) (Doc ID 2334800.1)

OAC-Classic : How to Setup Vanity URL and Custom SSL Certificate for OAC-Classic Instance Using LBaaS (Doc ID 2551524.1)

OAC (Oracle Managed) and OAC (Oracle Managed) on OCI we know that this feature is not yet supported and cannot setup a vanity URL to OAC

OAC - OM - How To Setup Vanity URL For Oracle Managed Instance (Doc ID 2606790.1)

There is an Enhancement request raised for this requirement.

Since the requirement is an Enhancement and does not exists as of now, what we discuss here in this document is a workaround that is a partial solution.

We cannot create a Vanity URL like (https://oacdev.companyname.com/dv/ui) for OAC that can be used as an alias URL for the Oracle provided OAC URL like

https://oacdev-tenancyname.analytics.opc.oraclecloud.com/dv/ui

or

https://oacdev-tenancyname.analytics.opc.oraclecloud.com/ui/dv

Even though you setup a DNS name and point to the existing Oracle Given URL hostname's IP address, there will be no are to setup the supported SSL Certificate and so upon accessing the DNS URL the browser throws certificate error.

In addition, the cloudgate at IDCS will not understand the custom Vanity URL's hostname and it will challenge for Authentication (401 Error) since it loses the session.

Therefore, we are proposing a workaround to setup an Apache HTTP Server either in Customer's network or in OCI Compute (Create an Oracle Linux Instance and install Apache) and configure through that Apache HTTP Server.

Important Note:

If we configure Apache as a Proxy/Reverse proxy Server, the same cloudgate issue will encounter since the cloudgate will not understand the new Vanity URL's hostname.

What will work?

Configure Apache Server as a WebServer on top of OAC and use Redirect using mod_rewrite module.

Upon End User entering the vanity URLin the browser it will redirect to IDCS for Login (If any External SAML IdP exists it will get redirected to External SAML SSO IdP) and after a successful authentication it gets redirected to the Oracle provided default OAC URL it will not stay on the vanity URL for further communication.

The Vanity URL is as if a link to the Oracle provided OAC URL, it is not a complete replacement for the Oracle provided OAC URL.

Who can apply this solution?

There are many end users who got used to the old OAC-Classic Vanity URL's and upon migration to OAC-Oracle Managed (OAC on OCI or OCI Native), it will be very difficult for the customer's Administrators to make their end users update the new OAC URL, then they can continue the same old vanity URL's pointing to the new OAC URL's using this solution.

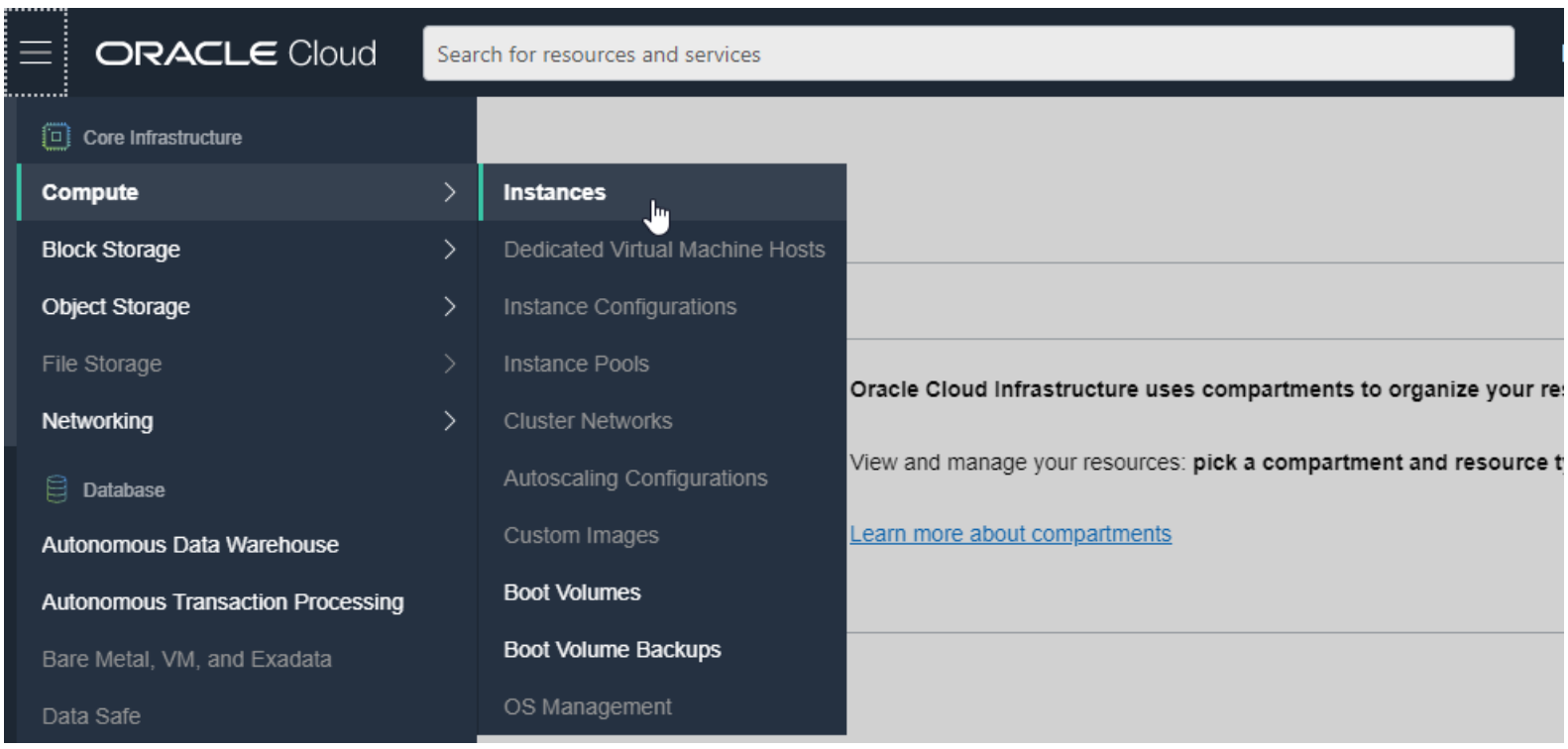
Steps to setup the solution is documented in this attachment.

Setup/Configuration steps:

Create an Oracle Linux/any Linux Instance, either in customer's network or in the OCI Compute

Install Apache on the Linux Instance.

If creating an OCI Compute Instance in OCI, we might need to work on the security configurations of the ingress and Egress Rules.



Apache:

Pre-Requisites: Install Apache HTTP Server with mod_proxy and mod_rewrite extensions/plugins enabled.

Will provide a separate document to Create Compute VM and install Apache HTTP Server in it.

Steps to Generate Private Key and CSR (To be done by Customer or any team in Oracle)

1. Logon to any Linux server using SSH access

2. Navigate to the /tmp directory: **cd /tmp**

3. Create a directory **/ssl** to hold the request file, the private key file, and the certificates if it does not already exist.:
mkdir ssl

4. Issue the following command:
>cd /tmp/ssl
>openssl req -new -newkey rsa:2048 -nodes -keyout /tmp/ssl/<key_file_name>.key -out /tmp/ssl/<cert_request_file_name>.csr

5. Follow the prompts by providing the requested values.

```
[veerao@den01dlv ~]$ openssl req -new -newkey rsa:2048 -nodes -keyout oacdev.key -out oacdev.csr
Generating a 2048 bit RSA private key
...................+++
....+++
writing new private key to 'oacdev.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:California
Locality Name (eg, city) [Default City]:RedwoodShores
Organization Name (eg, company) [Default Company Ltd]:your company name
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:oacdev.yourcompany.com
Email Address []:somebody@yourcompany.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
[veerao@den01dlv ~]$
```

6. The certs cannot be locked with a passphrase so simply press the **<Enter>** key for the passphrase question to get a blank passphrase. (See pre-reqs in Oracle doc on step 1 in next section.)

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
[veerao@den01dlv ~]$
```

7. This will generate the <key_file_name>.key file and the <cert_request_file_name>.csr file in the /tmp/ssl directory.

```
[veerao@den01dlv ~]$ ls -l oacdev*
-rw-r--r--+ 1 veerao dba 1082 Apr 16 04:53 oacdev.csr
-rw-r--r--+ 1 veerao dba 1704 Apr 16 04:53 oacdev.key
[veerao@den01dlv ~]$
```

8. Provide these two files for the certificate-signing request to the Customer's Network/Security Team who will buy the cert.

Get the CSR Signed by CA

Customer will buy the SSL Certificate from External CA for the DNS name

After CA signs the certificate request, Security Team should send back the certificates in .crt file format to OAC Team.

CA gives Signed OAC DNS Name SSL Certificate, CA Intermediate Certificate and CA Root Certificate.

Configure Apache with received SSL Certificates

1. Copy the received CA Signed Server certificate (oacdev.crt), CA Intermediate, CA Root certificates to /tmp/ssl
2. cat CA_Intermediate.crt and copy the content of the file shown as :
(-----BEGIN CERTIFICATE----- lkdoifjsad98w798ejr33u -----END CERTIFICATE-----)
3. cat CA_Root.crt and copy the content of the file shown as :
(-----BEGIN CERTIFICATE----- augsdwd6djksgds7d -----END CERTIFICATE-----)

4. vi server-ca.crt

5. First paste the copied CA_Intermediate.crt content (in step 2) to server-ca.crt

6. Go to the end of the content and in the next line append the copied content of CA_Root.crt (in step 3) to server-ca.crt

7. Save it, it should look like:

-----BEGIN CERTIFICATE-----

lkdoifjsad98w798ejr33u

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

augsdwd6djksgds7d

-----END CERTIFICATE-----

8. Copy the CA Signed Server certificate /tmp/ssl/oacdev.crt as /usr/local/apache2/conf/server.crt

9. Copy the Private key /tmp/ssl/oacdev.key as /usr/local/apache2/conf/server.key

10. Copy the intermediate chain cert /tmp/ssl/server-ca.crt to /usr/local/apache2/conf

11. cd /usr/local/apache2/conf

12. vi httpd.conf and set the ServerName from "xxxxxxxxxxx:80" to "oacdev.yourcompany.com:80"

13. cd /usr/local/apache2/conf/extra

14. vi httpd-ssl.conf

15. Change the SSLCipherSuite from "HIGH:MEDIUM:!MD5:!RC4:!3DES" to "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH"

16. Change the SSLProxyCipherSuite from "HIGH:MEDIUM:!MD5:!RC4:!3DES" to "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH"

17. Change the SSLProtocol from "all -SSLv3" to "TLSv1.2" (If you want only TLSv1.2) if not use "all -SSLv2 -SSLv3"

18. Change the SSLProxyProtocol from "all -SSLv3" to "TLSv1.2" (If you want only TLSv1.2) if not use "all -SSLv2 -SSLv3"

19. Get the CA Intermediate and CA Root certificates of target i.e OAC URL (https://oacdev.companyname.com/dv/ui) and create a /usr/local/apache2/conf/server-ca.crt file (CA Inter appended with CA Root)

20. Under the VirtualHost entry (<VirtualHost _default_:443>) change as below:

a. ServerName from "www.example.com:443" to oacdev.yourcompany.com:443"

b. Uncomment SSLCertificateFile "/usr/local/apache2/conf/server.crt"

c. Uncomment SSLCertificateKeyFile "/usr/local/apache2/conf/server.key"

d. Uncomment SSLCertificateChainFile "/usr/local/apache2/conf/server-ca.crt"

After Configuring SSL at Apache create a conf file redirect_http_to_https.conf under /usr/local/apache2/conf and Include it in the httpd.conf file.

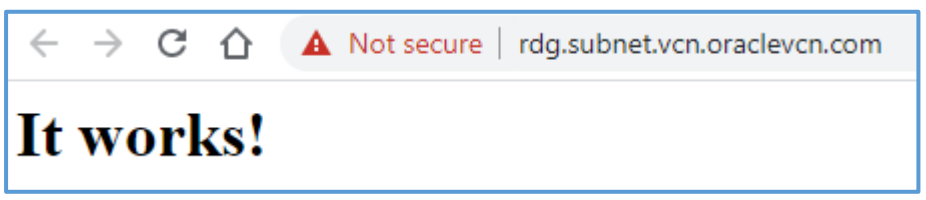
```
Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains"

<VirtualHost *:80>
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI}
</VirtualHost>
```

Open 80 and 443 ports on Apache Instance for Ingress at the VCN Security List of the Instance

Test the Apache URL is accessible (here the rdg.subnet.vcn.oraclevcn.com is the DNS Name given by the Compute instance's VCN)

This default DNS name is not a public accessible one, so you might need to get the Public IP of the VM and map the IP to the DNS name (oacdev.companyname.com) in both external and Internal DNS Servers.

https://rdg.subnet.vcn.oraclevcn.com/ → it shows "It Works!"



Now set the redirect

Create rewrite.conf under /usr/local/apache2/conf

NOTE: mod_rewrite module should be loaded

```
RewriteEngine On

Redirect Permanent "/" "https://devoac-[redacted]bo.analytics.ocp.oraclecloud.com/"
```

In the /usr/local/apache2/conf/httpd.conf file at the end Include the conf files.

Include conf/rewrite.conf

Include conf/redirect_http_to_https.conf

Restart apache services

Test

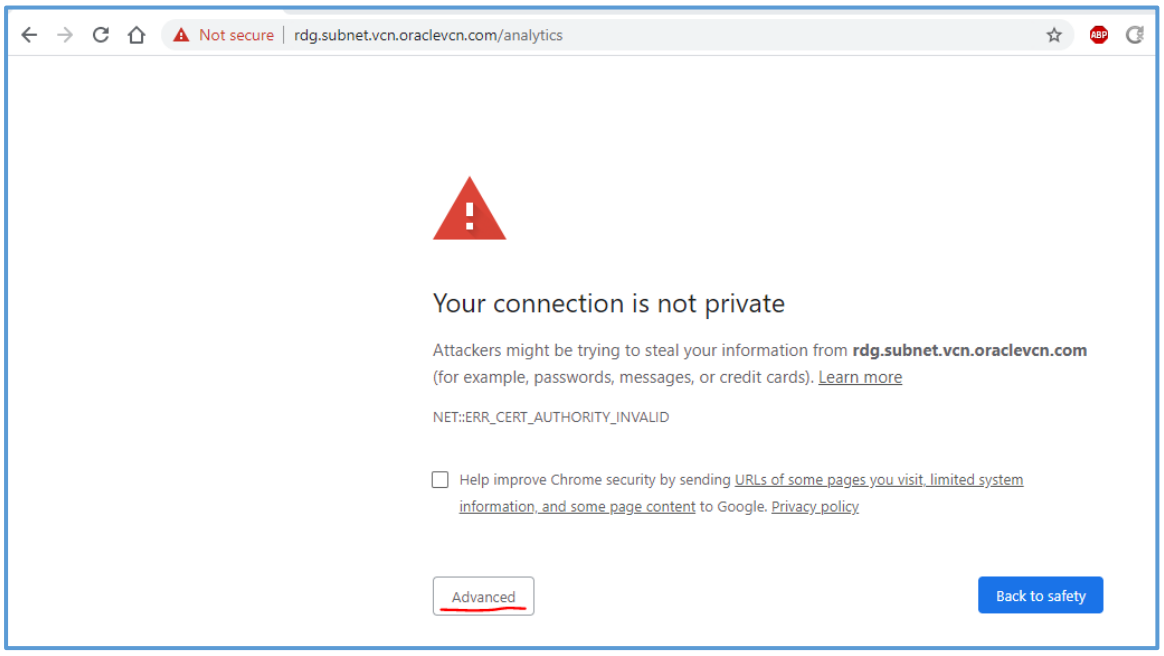
https://rdg.subnet.vcn.oraclevcn.com/analytics

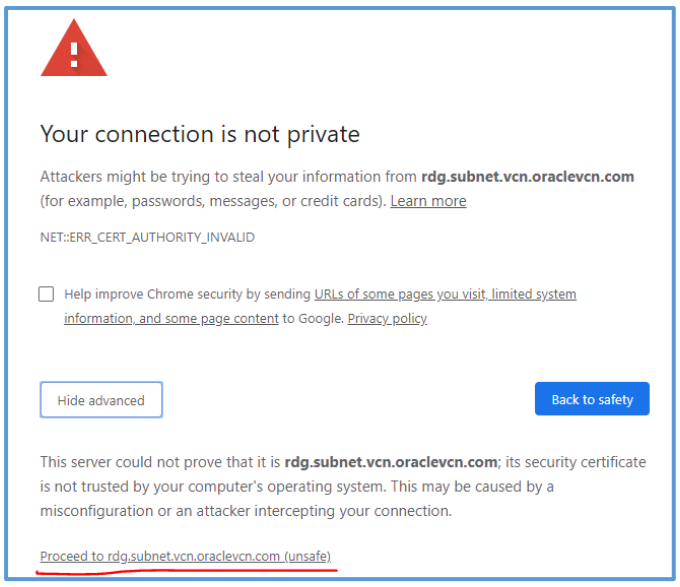
or

https://rdg.subnet.vcn.oraclevcn.com/ui/analytics (OAC on OCI Native)

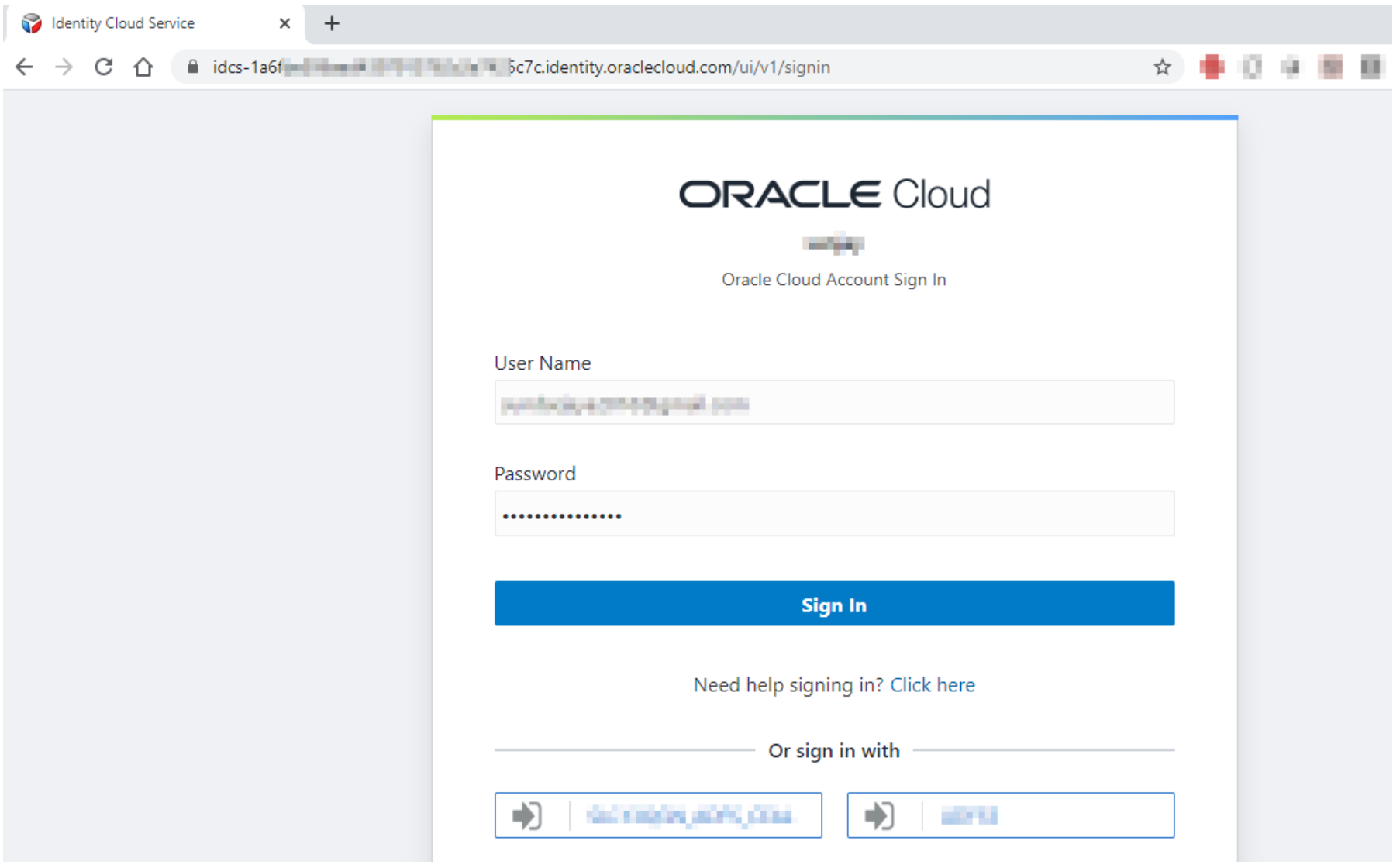
https://rdg.subnet.vcn.oraclevcn.com/ui/dv (OAC on OCI Native)

SSL Certificate exception screen is displayed since my SSL Certificate is self signed and a dummy certificate, you will not encounter this issue if your SSL certificate is a Public CA Signed certificate.

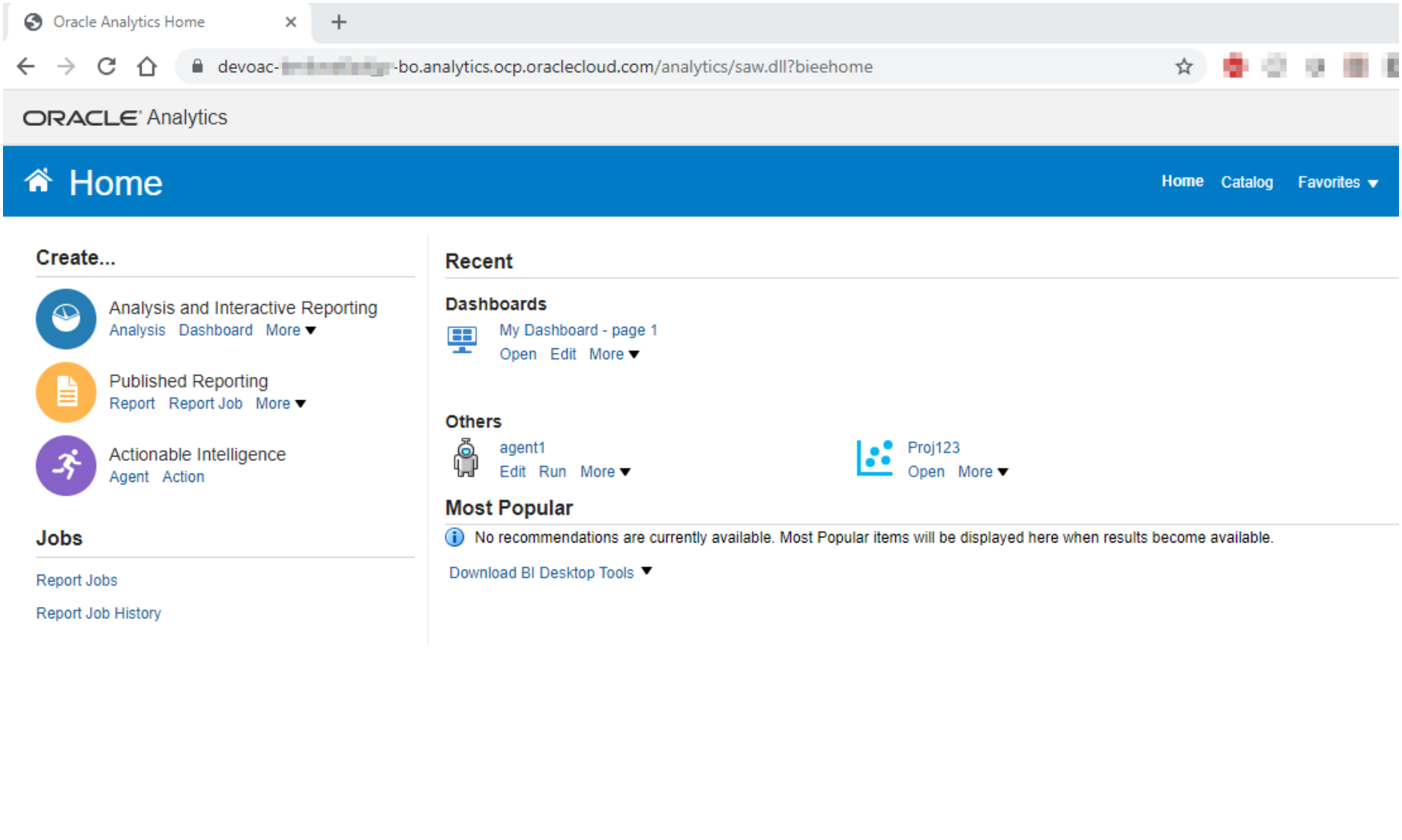


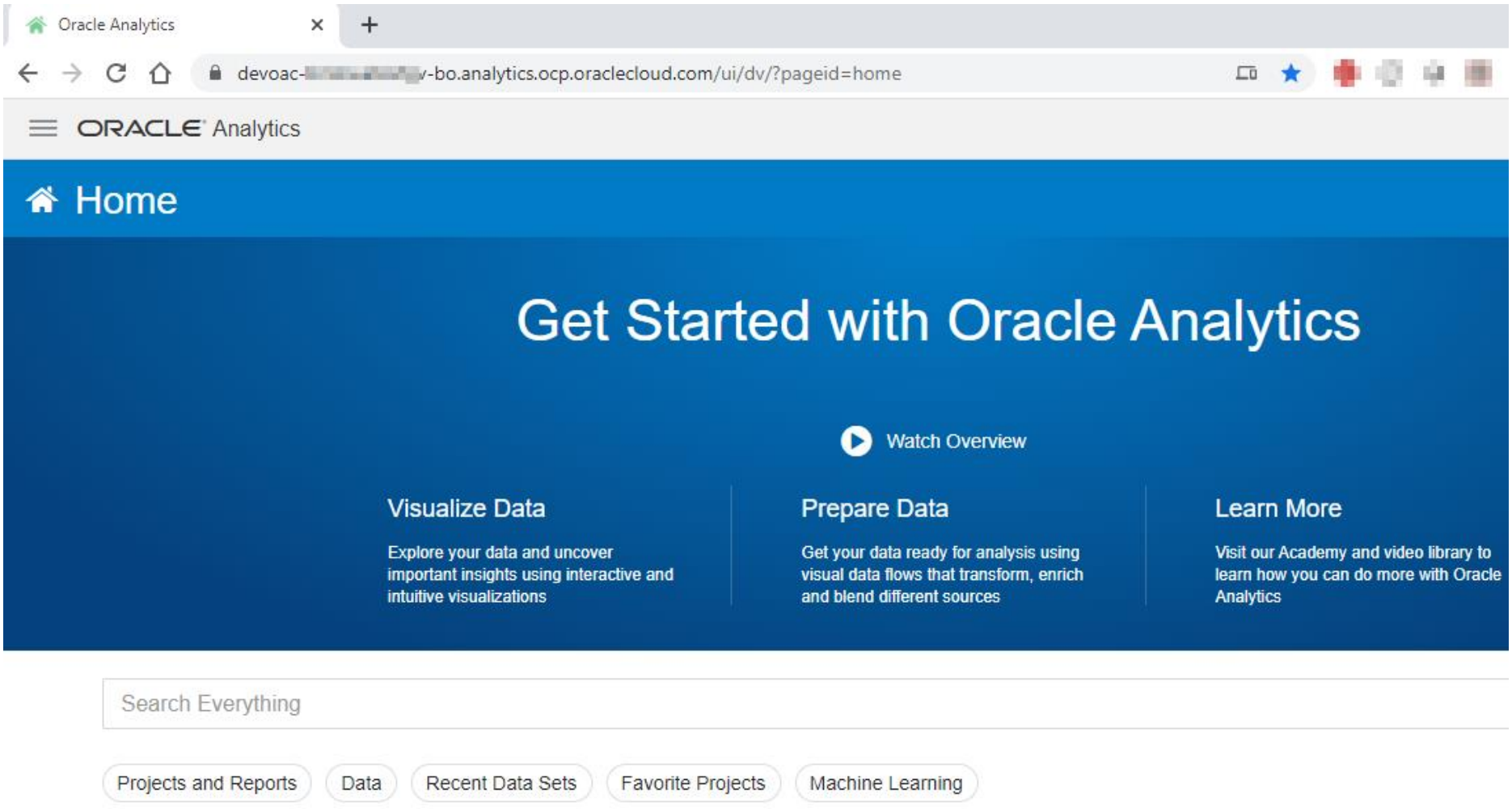


Gets redirected to IDCS Login page



After successful Authentication, it is redirected to the actual OAC URL (doesn't continue on the Vanity URL)




Oracle Analytics
devoac-[redacted]v-bo.analytics.ocp.oraclecloud.com/ui/dv/?pageid=home
ORACLE Analytics
Home
Get Started with Oracle Analytics
Watch Overview
Visualize Data
Explore your data and uncover important insights using interactive and intuitive visualizations
Prepare Data
Get your data ready for analysis using visual data flows that transform, enrich and blend different sources
Learn More
Visit our Academy and video library to learn how you can do more with Oracle Analytics
Search Everything
Projects and Reports
Data
Recent Data Sets
Favorite Projects
Machine Learning