# OPR Deployment Issues

We have categorized these issues into 3 buckets

# Tyndall On-Prem OPR Environment Issues - This the deployment at the client site at Tyndall AFB. OPR containers are being deployed on VMware Tanzu environment

1. BAH AWS Gov OPR environment - This the deployment at BAH AWS Gov Cloud. OPR containers are being deployed on AWS EKS cluster
2. BAH OPR packages integration with Tyndall DT app – This is task to integrate the OPR packages into the customized Reflect Tyndall DT application

Tyndall On-Prem OPR Environment Issues

Background: OPR containers are being deployed to a VMWare on-prem server. Tanzu is providing the Kubernetes layer. This deployment is being done at the client site at Tyndall AFB.

Issues (Row 9 in accompanying excel)

* Address OPR Helm Chart compatibility with Kubernetes 1.28
  + Update Horizontal Pod Autoscaler to use v2 API instead of v2beta2 (manually patched temporarily)
* Fix issues with Monitoring CRDs
  + Replace ServiceMonitor resources which require Prometheus Operator (deployed Monitoring from BigBang to resolve Prometheus Operator dependency)
* Rewrite dependency on ingress-nginx to use Istio VirtualService
* Fix PostgresQL helm syntax for {}.username in OPR helm chart
  + Disabled PostgresQL in values.yaml file as a workaround

BAH AWS Gov OPR environment

Background: OPR containers are being deployed to a AWS Gov Cloud env owned by BAH.

Issues(Row 1 in accompanying excel)

Reflect-sync reboot loops with the following exit error.

[15:59:36 ERR] An error occurred using the connection to database 'ProjectService' on server ''. <s:Microsoft.EntityFrameworkCore.Database.Connection>

The postgres database pod is running.

Notice that in the POSTGRES variables it is filling in to a 172.10 address which does not match the VLAN range within the EKS cluster. This cannot be changed within AWS with the VPC CNI plugin and restrictions to allowed ranges with our enterprise team.

Top level output:

[15:59:35 DBG] Hosting starting <s:Microsoft.Extensions.Hosting.Internal.Host>

[15:59:35 WRN] No XML encryptor configured. Key {0624efa5-3830-44d1-81d5-83d50d20d856} may be persisted to storage in unencrypted form. <s:Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager>

[15:59:35 INF] Starting Prometheus Exporter on port 1234 <s:Unity.Opr.ProjectServer.Server.Grpc.ServerHost>

[15:59:35 INF] Grpc Server starting [0.0.0.0:10010] <s:Unity.Opr.ProjectServer.Server.Grpc.ServerHost>

[15:59:35 WRN] 'AddEntityFramework\*' was called on the service provider, but 'UseInternalServiceProvider' wasn't called in the DbContext options configuration. Consider removing the 'AddEntityFramework\*' call, as in most cases it's not needed and may cause conflicts with other products and services registered in the same service provider. <s:Microsoft.EntityFrameworkCore.Infrastructure>

[15:59:36 WRN] Sensitive data logging is enabled. Log entries and exception messages may include sensitive application data; this mode should only be enabled during development. <s:Microsoft.EntityFrameworkCore.Model.Validation>

[15:59:36 WRN] The foreign key property 'DeployIdPViewerBundleId.DeployIdPId1' was created in shadow state because a conflicting property with the simple name 'DeployIdPId' exists in the entity type, but is either not mapped, is already used for another relationship, or is incompatible with the associated primary key type. See <https://aka.ms/efcore-relationships> for information on mapping relationships in EF Core. <s:Microsoft.EntityFrameworkCore.Model.Validation>

[15:59:36 WRN] The 'bool' property 'IsOnline' on entity type 'SyncServer' is configured with a database-generated default. This default will always be used for inserts when the property has the value 'false', since this is the CLR default for the 'bool' type. Consider using the nullable 'bool?' type instead, so that the default will only be used for inserts when the property value is 'null'. <s:Microsoft.EntityFrameworkCore.Model.Validation>

[15:59:36 ERR] An error occurred using the connection to database 'ProjectService' on server ''. <s:Microsoft.EntityFrameworkCore.Database.Connection>

[15:59:36 FTL] Unexpected exception during floating seat database cleanup <s:Unity.Opr.ProjectServer.API.Data.DataCleanupService>

System.Net.Internals.SocketExceptionFactory+ExtendedSocketException (00000005, 0xFFFDFFFF): Name or service not known

at System.Net.Dns.GetHostEntryOrAddressesCore(String hostName, Boolean justAddresses, AddressFamily addressFamily, ValueStopwatch stopwatch)

at System.Net.Dns.<>c.<GetHostEntryOrAddressesCoreAsync>b\_\_33\_0(Object s, ValueStopwatch stopwatch)

at System.Net.Dns.<>c\_\_DisplayClass39\_0`1.<RunAsync>b\_\_0(Task <p0>, Object <p1>)

at System.Threading.Tasks.ContinuationResultTaskFromTask`1.InnerInvoke()

at System.Threading.Tasks.Task.<>c.<.cctor>b\_\_272\_0(Object obj)

at System.Threading.ExecutionContext.RunFromThreadPoolDispatchLoop(Thread threadPoolThread, ExecutionContext executionContext, ContextCallback callback, Object state)

--- End of stack trace from previous location ---

This issue presents several red flags based on the logs, and it can be approached systematically to troubleshoot and resolve the problem.

**Key Issues and Analysis**

1. **PostgreSQL Connection Error:**
   * **Error Message:** [15:59:36 ERR] An error occurred using the connection to database 'ProjectService' on server ''.
   * This indicates that the connection to the PostgreSQL database is failing. The key point is that the server name is missing (server ''), which means there’s a problem with how the PostgreSQL connection string is being populated.
2. **PostgreSQL Pod and Networking Issue:**
   * The log mentions that the PostgreSQL pod is running, but the POSTGRES variables are being filled in with a 172.10.x.x address. This address does not match the expected VLAN range within the EKS cluster.
   * The IP address being assigned to the PostgreSQL pod seems to fall outside the range allowed by the enterprise networking restrictions. This suggests that the EKS CNI (Container Network Interface) plugin configuration or the VPC subnet allocation might not be properly aligned with the cluster's networking policies.
3. **Entity Framework Core Warnings:**
   * Multiple warnings related to Entity Framework Core are present:
     + 'AddEntityFramework\*' was called but 'UseInternalServiceProvider' wasn't used.
     + Sensitive data logging is enabled.
     + Foreign key property conflicts.
     + IsOnline bool property and default value conflicts.
   * While these warnings may not be the root cause, they point to potential issues in the database model configuration and should be addressed after resolving the main issue.
4. **Socket Exception:**
   * **Error Message:** System.Net.Internals.SocketExceptionFactory+ExtendedSocketException: Name or service not known.
   * This indicates that the app is unable to resolve the database service hostname, further reinforcing that there’s a problem with the connection string and possibly the DNS resolution for the PostgreSQL service.

**Approach to Troubleshooting and Resolution**

**1. Check and Validate PostgreSQL Environment Variables:**

* **Issue:** The PostgreSQL variables are incorrectly configured with an IP address (172.10.x.x) that does not belong to the allowed VLAN range within the EKS cluster.
* **Action:**
  + Verify the POSTGRES\_HOST, POSTGRES\_USER, POSTGRES\_PASSWORD, and POSTGRES\_DB environment variables set for the application.
  + Confirm that the PostgreSQL service is using a valid cluster IP or hostname that can be reached by the app. In Kubernetes, this typically should be a service name (postgres-service.default.svc.cluster.local) rather than a specific IP.
  + Update the connection string to use the correct DNS name or service name of the PostgreSQL service.

**2. Investigate the Networking Setup in EKS:**

* **Issue:** The PostgreSQL pod is receiving an IP address (172.10.x.x) outside the allowed VLAN range.
* **Action:**
  + Check the EKS CNI plugin configuration. Ensure that the VPC and subnet CIDR ranges are correctly aligned with your enterprise networking policies.
  + Inspect the subnet configurations within the VPC to confirm they fall within the expected ranges.
  + You can use kubectl get pods -o wide to confirm the IP addresses assigned to the pods and verify the network configuration.

**3. Review and Fix Entity Framework Core Configuration:**

* **Issue:** Several warnings in the logs relate to improper EF Core configuration, such as:
  + 'AddEntityFramework\*' was called without 'UseInternalServiceProvider'.
  + Sensitive data logging is enabled (should be disabled in production).
  + Foreign key and default value configuration issues.
* **Action:**
  + Check the DbContext configuration and ensure that UseInternalServiceProvider is properly called if needed.
  + Disable sensitive data logging in production environments by modifying the DbContextOptionsBuilder in the EF Core configuration (options.EnableSensitiveDataLogging(false)).
  + Investigate the foreign key and default value issues mentioned in the warnings and apply the necessary fixes to the entity models.

**4. DNS Resolution and Service Discovery:**

* **Issue:** The application cannot resolve the PostgreSQL service hostname (Name or service not known error).
* **Action:**
  + Ensure that the PostgreSQL service is correctly exposed in Kubernetes, either via a ClusterIP or Headless service, depending on the deployment needs.
  + Validate that DNS resolution is working correctly within the Kubernetes cluster. You can test this by running a simple pod (busybox or alpine) and using tools like nslookup or dig to resolve the PostgreSQL service.
  + Check that CoreDNS is functioning properly in the EKS cluster (kubectl get pods -n kube-system to inspect CoreDNS pods).

**5. Check Pod Logs and Retry Mechanism:**

* **Issue:** Reflect-sync seems to be in a reboot loop, which might be related to repeated connection attempts to the PostgreSQL service.
* **Action:**
  + Investigate the logs further to check if the pod is being restarted frequently due to liveness or readiness probe failures.
  + If necessary, implement a retry mechanism in the application to handle transient connection failures gracefully.

**Conclusion**

The core issue seems to stem from a misconfiguration in the PostgreSQL connection string or an issue with the EKS network configuration that causes the app to receive an invalid IP address for the database. Start by fixing the PostgreSQL environment variables and ensuring the correct networking setup in EKS. Once the connectivity issues are resolved, focus on optimizing the EF Core configuration and addressing the warnings.

Let me know if you'd like more detailed guidance on any of the steps!

`````````````````````````````````````````````````````````````````````

To troubleshoot the PostgreSQL pod and networking issue where the 172.10.x.x address assigned to the PostgreSQL pod falls outside the expected VLAN range in your EKS cluster, follow this step-by-step process:

**1. Verify the Network Configuration of the PostgreSQL Pod**

* **Check Pod's IP Address and Network Assignment:** Use the following command to get the detailed information about the PostgreSQL pod, including the IP address, assigned node, and namespace:

bash

Copy code

kubectl get pod <postgresql-pod-name> -o wide

This will show the pod’s assigned IP, node, and subnet. Verify if the IP (172.10.x.x) falls outside the expected CIDR range for your VPC subnets.

* + **Expected Range:** Compare the IP address range assigned to the pod with the VPC’s CIDR blocks, which you can find in the AWS console under VPC configuration.

**2. Verify the Kubernetes Networking (CNI) Configuration**

* **Check EKS CNI Plugin Version:** Make sure the Amazon VPC CNI plugin is correctly installed and up to date. You can check the version running in your cluster with:

bash

Copy code

kubectl describe daemonset aws-node -n kube-system

The output should provide the image and version of the CNI plugin. Make sure it's the latest compatible version with your EKS cluster.

* **Check CNI Plugin Logs:** Look for any errors or warnings in the CNI logs that might indicate a problem with network allocation:

bash

Copy code

kubectl logs -n kube-system daemonset/aws-node

* **Check the IP Address Allocation:** Use the ipamd utility provided by AWS for troubleshooting IP address allocation issues. It shows the current IP addresses allocated to the pods. You can check it by looking at the aws-node pod logs:

bash

Copy code

kubectl logs -n kube-system <aws-node-pod-name>

**3. Verify Subnet and VPC CIDR Block Configurations**

* **Check VPC and Subnets CIDR Ranges:** In the AWS console, navigate to the VPC section and review the following:
  + VPC CIDR blocks.
  + Subnet CIDR blocks (both public and private).

Ensure that the assigned IP addresses for pods fall within the CIDR blocks allocated to the EKS cluster subnets.

* **Misaligned CIDR Block:** If the 172.10.x.x address doesn’t match your expected range, it’s likely that the VPC or subnets CIDR blocks were not properly aligned during cluster setup. You might need to recreate or adjust the subnets.
* **VLAN/Enterprise Networking Policy:** If there are networking restrictions (e.g., enterprise VLANs or IP blocks not allowed), confirm that the VPC’s CIDR block aligns with those requirements. You can work with your enterprise networking team to verify this.

**4. Review and Update VPC CNI Configuration**

* **Increase IP Address Availability:** If the IP addresses from the VPC subnets are exhausted, Kubernetes will assign addresses outside the expected range. To check for IP exhaustion:

bash

Copy code

kubectl describe node <node-name>

Look at the available IPs and compare with the total assigned. If you're running low on available IPs, you may need to:

* + Increase the size of the subnets or add more subnets.
  + Use the ENIConfig feature if you're using multiple VPCs or subnets to ensure correct assignment of IPs.
* **CNI Custom Networking:** If you have customized CNI networking, review the configuration for any issues with subnet allocation. AWS allows specifying which subnets to use for pod IPs through CNI custom networking.
* **Adjust the CNI Config:** You can modify the aws-node CNI configuration for specific requirements. This might involve setting specific subnet allocations or limiting the ranges used for pod IPs. You can check the configuration map with:

bash

Copy code

kubectl describe configmap aws-node -n kube-system

**5. Validate DNS and Service Discovery**

If the PostgreSQL pod has the correct IP but the app is still unable to reach it, there might be DNS issues within the cluster.

* **Check Service Configuration:** Ensure that the PostgreSQL service (ClusterIP, Headless) is properly configured to expose the database to other pods. Use the following command to review the service:

bash

Copy code

kubectl get svc <postgresql-service-name> -o yaml

* **DNS Resolution:** Use a debug pod (e.g., busybox or alpine) to check if DNS resolution works correctly for the PostgreSQL service. For example:

bash

Copy code

kubectl run -i --tty dns-test --image=busybox --restart=Never -- sh

nslookup <postgresql-service-name>

**6. Review VPC Security Group and NACL Rules**

Check if there are any restrictive security group or NACL (Network Access Control List) rules that prevent communication between the pods and the PostgreSQL pod.

* **VPC Security Groups:** Ensure that the PostgreSQL pod and the application pod are within security groups that allow traffic on the necessary ports (e.g., 5432 for PostgreSQL).
* **Network ACLs:** Verify that the Network ACLs for the subnets allow traffic within the necessary port ranges between the pods in the EKS cluster.

**7. Test with a Standalone Postgres Client**

If the networking setup is suspected to be the issue, deploy a simple PostgreSQL client (such as psql from a container) in the cluster to attempt direct connections to the PostgreSQL pod:

bash

Copy code

kubectl run -i --tty psql-client --image=postgres --restart=Never -- sh

psql -h <postgresql-service-name> -U <username> -d <dbname>

This will help you isolate whether the problem is with the app itself or with the underlying network.

**Conclusion**

The problem is likely related to either a mismatch in the CIDR blocks used in the VPC or IP exhaustion in the subnets. By systematically checking the pod's assigned IP, verifying the VPC and subnet configurations, and ensuring the CNI plugin is correctly set up, you should be able to identify the root cause and resolve the issue.

Let me know if you need further assistance or more detailed commands for any of these steps!

BAH OPR packages integration with Tyndall DT app

Background: We have a customized Reflect to build the Tyndall Digital Twin application. We are in the process of integrating the OPR packages with our digital Twin app. Facing the following issues during these

Issues

BAH AWS login troubleshooting

* Some clarity on the OPR login process specifically how the login manager, ui state manager and interop interact with each other.
* Better error handling , confusing to track issues when licensing stuff, server changes, ip address/host file issues. The server won’t connect and there won’t be any exceptions thrown, and debug.logs look almost identical

Licensing server issues

* More clarity on how the licensing server works with opr
* We’ve had a licensing error appear when switching over to unity industry, but it popped up infrequently:  
  Direct stack trace for that error:  
    
  9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ValidateEnd (System.Threading.Tasks.Task task) [0x00008] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1+ConfiguredTaskAwaiter[TResult].GetResult () [0x00000] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at Unity.Opr.ProjectServerHttpClient+<RequestAsync>d\_\_26`1[T].MoveNext () [0x000e9] in <da36b501918c4619851d20f3926cf44c>:0   
   --- End of stack trace from previous location where exception was thrown ---  
   at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw () [0x0000c] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess (System.Threading.Tasks.Task task) [0x0003e] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification (System.Threading.Tasks.Task task) [0x00028] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ValidateEnd (System.Threading.Tasks.Task task) [0x00008] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1+ConfiguredTaskAwaiter[TResult].GetResult () [0x00000] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at Unity.Opr.ProjectServerHttpClient+<GetAsync>d\_\_25`1[T].MoveNext () [0x0007d] in <da36b501918c4619851d20f3926cf44c>:0   
   --- End of stack trace from previous location where exception was thrown ---  
   at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw () [0x0000c] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess (System.Threading.Tasks.Task task) [0x0003e] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification (System.Threading.Tasks.Task task) [0x00028] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ValidateEnd (System.Threading.Tasks.Task task) [0x00008] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1+ConfiguredTaskAwaiter[TResult].GetResult () [0x00000] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at Unity.Opr.ProjectServerHttpClient+<GetUserInfoCore>d\_\_2.MoveNext () [0x00077] in <da36b501918c4619851d20f3926cf44c>:0   
   --- End of stack trace from previous location where exception was thrown ---  
   at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw () [0x0000c] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess (System.Threading.Tasks.Task task) [0x0003e] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification (System.Threading.Tasks.Task task) [0x00028] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ValidateEnd (System.Threading.Tasks.Task task) [0x00008] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1+ConfiguredTaskAwaiter[TResult].GetResult () [0x00000] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at Unity.Opr.ProjectServerClient+<GetUserInfo>d\_\_38.MoveNext () [0x000ac] in <da36b501918c4619851d20f3926cf44c>:0   
   --- End of inner exception stack trace ---  
   ---> (Inner Exception #0) Unity.Opr.Utils.Errors.UnauthorizedException  
   at Unity.Opr.Utils.ReflectHttpUtils+<GetResponse>d\_\_0.MoveNext () [0x00172] in <d1fe4648b71c42dcaaf10f0f13614d1c>:0   
   --- End of stack trace from previous location where exception was thrown ---  
   at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw () [0x0000c] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess (System.Threading.Tasks.Task task) [0x0003e] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification (System.Threading.Tasks.Task task) [0x00028] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ValidateEnd (System.Threading.Tasks.Task task) [0x00008] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1+ConfiguredTaskAwaiter[TResult].GetResult () [0x00000] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at Unity.Opr.Utils.ReflectHttpUtils+<GetResponse>d\_\_1`1[T].MoveNext () [0x00080] in <d1fe4648b71c42dcaaf10f0f13614d1c>:0   
   --- End of stack trace from previous location where exception was thrown ---  
   at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw () [0x0000c] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess (System.Threading.Tasks.Task task) [0x0003e] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification (System.Threading.Tasks.Task task) [0x00028] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ValidateEnd (System.Threading.Tasks.Task task) [0x00008] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1+ConfiguredTaskAwaiter[TResult].GetResult () [0x00000] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at Unity.Opr.ProjectServerHttpClient+<RequestAsync>d\_\_26`1[T].MoveNext () [0x000e9] in <da36b501918c4619851d20f3926cf44c>:0   
   --- End of stack trace from previous location where exception was thrown ---  
   at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw () [0x0000c] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess (System.Threading.Tasks.Task task) [0x0003e] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification (System.Threading.Tasks.Task task) [0x00028] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ValidateEnd (System.Threading.Tasks.Task task) [0x00008] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1+ConfiguredTaskAwaiter[TResult].GetResult () [0x00000] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at Unity.Opr.ProjectServerHttpClient+<GetAsync>d\_\_25`1[T].MoveNext () [0x0007d] in <da36b501918c4619851d20f3926cf44c>:0   
   --- End of stack trace from previous location where exception was thrown ---  
   at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw () [0x0000c] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess (System.Threading.Tasks.Task task) [0x0003e] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification (System.Threading.Tasks.Task task) [0x00028] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ValidateEnd (System.Threading.Tasks.Task task) [0x00008] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1+ConfiguredTaskAwaiter[TResult].GetResult () [0x00000] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at Unity.Opr.ProjectServerHttpClient+<GetUserInfoCore>d\_\_2.MoveNext () [0x00077] in <da36b501918c4619851d20f3926cf44c>:0   
   --- End of stack trace from previous location where exception was thrown ---  
   at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw () [0x0000c] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess (System.Threading.Tasks.Task task) [0x0003e] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification (System.Threading.Tasks.Task task) [0x00028] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.TaskAwaiter.ValidateEnd (System.Threading.Tasks.Task task) [0x00008] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at System.Runtime.CompilerServices.ConfiguredTaskAwaitable`1+ConfiguredTaskAwaiter[TResult].GetResult () [0x00000] in <695d1cc93cca45069c528c15c9fdd749>:0   
   at Unity.Opr.ProjectServerClient+<GetUserInfo>d\_\_38.MoveNext () [0x000ac] in <da36b501918c4619851d20f3926cf44c>:0 <---

UnityEngine.SetupCoroutine:InvokeMoveNext (System.Collections.IEnumerator,intptr)

Fix issues with Reflect Viewer hardcoded as directory name (lower priority):  
stack trace:ArgumentOutOfRangeException: Length cannot be less than zero.  
Parameter name: length  
System.String.Substring (System.Int32 startIndex, System.Int32 length) (at <695d1cc93cca45069c528c15c9fdd749>:0)  
UnityEngine.Reflect.AuthBackend.GetResolverPath () (at C:/Users/626555/Documents/GitHub/Tyndall\_Base/Packages/com.unity.opr/Runtime/Utils/Editor/Windows/AuthBackend.cs:70)  
UnityEngine.Reflect.AuthBackend.AddRegistryKeys () (at C:/Users/626555/Documents/GitHub/Tyndall\_Base/Packages/com.unity.opr/Runtime/Utils/Editor/Windows/AuthBackend.cs:80)  
UnityEngine.Reflect.AuthBackend.Login () (at C:/Users/626555/Documents/GitHub/Tyndall\_Base/Packages/com.unity.opr/Runtime/Utils/Editor/Windows/AuthBackend.cs:38)  
UnityEngine.Reflect.LoginManager.Login () (at C:/Users/626555/Documents/GitHub/Tyndall\_Base/Packages/com.unity.opr/Runtime/Utils/LoginManager.cs:709)  
Unity.Reflect.Viewer.UI.UIStateManager+<StartLogin>d\_\_291.MoveNext () (at Assets/Unity Reflect Assets/Scripts/UI/UIStateManagerPipeline.cs:1072)  
UnityEngine.SetupCoroutine.InvokeMoveNext (System.Collections.IEnumerator enumerator, System.IntPtr returnValueAddress) (at <edb09cd490414b2684a18fe60d255a59>:0)the error seems to crash at this line: var subPath = appDomainLocation.Substring(0, appDomainLocation.LastIndexOf("ReflectViewer")); which fails because the project directory doesn't have reflectviewer in it.