* Code

I am deploying some applications n my eks cluster, and I am having this issue::: Configuration values

Environment: Development

AllowedHosts = \*

ANNOTATIONS\_PORT = tcp://172.20.14.69:80

ANNOTATIONS\_PORT\_1234\_TCP = tcp://172.20.14.69:1234

ANNOTATIONS\_PORT\_1234\_TCP\_ADDR = 172.20.14.69

ANNOTATIONS\_PORT\_1234\_TCP\_PORT = 1234

ANNOTATIONS\_PORT\_1234\_TCP\_PROTO = tcp

ANNOTATIONS\_PORT\_80\_TCP = tcp://172.20.14.69:80

ANNOTATIONS\_PORT\_80\_TCP\_ADDR = 172.20.14.69

ANNOTATIONS\_PORT\_80\_TCP\_PORT = 80

ANNOTATIONS\_PORT\_80\_TCP\_PROTO = tcp

ANNOTATIONS\_SERVICE\_HOST = 172.20.14.69

ANNOTATIONS\_SERVICE\_PORT = 80

ANNOTATIONS\_SERVICE\_PORT\_HTTP = 80

ANNOTATIONS\_SERVICE\_PORT\_PROM\_PUBLISHER = 1234

applicationName = Unity.Opr.ProjectServer

ASPNETCORE\_URLS = http://+:80

COMMONUI\_PORT = tcp://172.20.180.10:5000

COMMONUI\_PORT\_5000\_TCP = tcp://172.20.180.10:5000

COMMONUI\_PORT\_5000\_TCP\_ADDR = 172.20.180.10

COMMONUI\_PORT\_5000\_TCP\_PORT = 5000

COMMONUI\_PORT\_5000\_TCP\_PROTO = tcp

COMMONUI\_SERVICE\_HOST = 172.20.180.10

COMMONUI\_SERVICE\_PORT = 5000

COMMONUI\_SERVICE\_PORT\_HTTP = 5000

ConnectionStrings:ProjectServiceDatabase = Host=unity.project-db.local;Database=ProjectService;Username=project\_service;Password=reflect

container = oci

contentRoot = /app

DashboardDomains:0 = https://dashboard.reflect.unity3d.com

development-certificate.pfx = -----BEGIN CERTIFICATE-----

MIIDPDCCAiSgAwIBAgIQX6hlMY3w67BBZyntLICtwjANBgkqhkiG9w0BAQsFADAg

MR4wHAYD,,,,,,,,,w==

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

environment = Development

GIT\_COMMIT =

HOME = /home/dotnet

HOSTNAME = reflect-project

HttpServer:ExceptionContextResponseEnabled = False

KUBERNETES\_PORT = tcp://172.20.0.1:443

KUBERNETES\_PORT\_443\_TCP = tcp://172.20.0.1:443

KUBERNETES\_PORT\_443\_TCP\_ADDR = 172.20.0.1

KUBERNETES\_PORT\_443\_TCP\_PORT = 443

KUBERNETES\_PORT\_443\_TCP\_PROTO = tcp

KUBERNETES\_SERVICE\_HOST = 172.20.0.1

KUBERNETES\_SERVICE\_PORT = 443

KUBERNETES\_SERVICE\_PORT\_HTTPS = 443

LINKS\_PORT = tcp://172.20.117.171:80

LINKS\_PORT\_80\_TCP = tcp://172.20.117.171:80

LINKS\_PORT\_80\_TCP\_ADDR = 172.20.117.171

LINKS\_PORT\_80\_TCP\_PORT = 80

LINKS\_PORT\_80\_TCP\_PROTO = tcp

LINKS\_SERVICE\_HOST = 172.20.117.171

LINKS\_SERVICE\_PORT = 80

LINKS\_SERVICE\_PORT\_HTTP = 80

LinkSharing:Domain = https://reflect.unity3d.com/

LinkSharing:Path = p/

OpenTelemetry:Enabled = True

OpenTelemetry:UseExporter = Console

OPR\_REFLECT\_POSTGRESQL\_PORT = tcp://172.20.81.195:5432

OPR\_REFLECT\_POSTGRESQL\_PORT\_5432\_TCP = tcp://172.20.81.195:5432

OPR\_REFLECT\_POSTGRESQL\_PORT\_5432\_TCP\_ADDR = 172.20.81.195

OPR\_REFLECT\_POSTGRESQL\_PORT\_5432\_TCP\_PORT = 5432

OPR\_REFLECT\_POSTGRESQL\_PORT\_5432\_TCP\_PROTO = tcp

OPR\_REFLECT\_POSTGRESQL\_SERVICE\_HOST = 172.20.81.195

OPR\_REFLECT\_POSTGRESQL\_SERVICE\_PORT = 5432

OPR\_REFLECT\_POSTGRESQL\_SERVICE\_PORT\_TCP\_POSTGRESQL = 5432

OPR\_REFLECT\_RABBITMQ\_PORT = tcp://172.20.70.78:5672

OPR\_REFLECT\_RABBITMQ\_PORT\_15672\_TCP = tcp://172.20.70.78:15672

OPR\_REFLECT\_RABBITMQ\_PORT\_15672\_TCP\_ADDR = 172.20.70.78

OPR\_REFLECT\_RABBITMQ\_PORT\_15672\_TCP\_PORT = 15672

OPR\_REFLECT\_RABBITMQ\_PORT\_15672\_TCP\_PROTO = tcp

OPR\_REFLECT\_RABBITMQ\_PORT\_25672\_TCP = tcp://172.20.70.78:25672

OPR\_REFLECT\_RABBITMQ\_PORT\_25672\_TCP\_ADDR = 172.20.70.78

OPR\_REFLECT\_RABBITMQ\_PORT\_25672\_TCP\_PORT = 25672

OPR\_REFLECT\_RABBITMQ\_PORT\_25672\_TCP\_PROTO = tcp

OPR\_REFLECT\_RABBITMQ\_PORT\_4369\_TCP = tcp://172.20.70.78:4369

OPR\_REFLECT\_RABBITMQ\_PORT\_4369\_TCP\_ADDR = 172.20.70.78

OPR\_REFLECT\_RABBITMQ\_PORT\_4369\_TCP\_PORT = 4369

OPR\_REFLECT\_RABBITMQ\_PORT\_4369\_TCP\_PROTO = tcp

OPR\_REFLECT\_RABBITMQ\_PORT\_5672\_TCP = tcp://172.20.70.78:5672

OPR\_REFLECT\_RABBITMQ\_PORT\_5672\_TCP\_ADDR = 172.20.70.78

OPR\_REFLECT\_RABBITMQ\_PORT\_5672\_TCP\_PORT = 5672

OPR\_REFLECT\_RABBITMQ\_PORT\_5672\_TCP\_PROTO = tcp

OPR\_REFLECT\_RABBITMQ\_SERVICE\_HOST = 172.20.70.78

OPR\_REFLECT\_RABBITMQ\_SERVICE\_PORT = 5672

OPR\_REFLECT\_RABBITMQ\_SERVICE\_PORT\_AMQP = 5672

OPR\_REFLECT\_RABBITMQ\_SERVICE\_PORT\_DIST = 25672

OPR\_REFLECT\_RABBITMQ\_SERVICE\_PORT\_EPMD = 4369

OPR\_REFLECT\_RABBITMQ\_SERVICE\_PORT\_HTTP\_STATS = 15672

PATH = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

POSTGRES\_INTEGRATION\_SVC\_PORT = tcp://172.20.143.12:5432

POSTGRES\_INTEGRATION\_SVC\_PORT\_5432\_TCP = tcp://172.20.143.12:5432

POSTGRES\_INTEGRATION\_SVC\_PORT\_5432\_TCP\_ADDR = 172.20.143.12

POSTGRES\_INTEGRATION\_SVC\_PORT\_5432\_TCP\_PORT = 5432

POSTGRES\_INTEGRATION\_SVC\_PORT\_5432\_TCP\_PROTO = tcp

POSTGRES\_INTEGRATION\_SVC\_SERVICE\_HOST = 172.20.143.12

POSTGRES\_INTEGRATION\_SVC\_SERVICE\_PORT = 5432

POSTGRES\_INTEGRATION\_SVC\_SERVICE\_PORT\_POSTRGRES\_INTEGRATION\_TEST\_SVC = 5432

PROJECT\_GRPC\_PORT = tcp://172.20.55.3:10010

PROJECT\_GRPC\_PORT\_10010\_TCP = tcp://172.20.55.3:10010

PROJECT\_GRPC\_PORT\_10010\_TCP\_ADDR = 172.20.55.3

PROJECT\_GRPC\_PORT\_10010\_TCP\_PORT = 10010

PROJECT\_GRPC\_PORT\_10010\_TCP\_PROTO = tcp

PROJECT\_GRPC\_SERVICE\_HOST = 172.20.55.3

PROJECT\_GRPC\_SERVICE\_PORT = 10010

PROJECT\_GRPC\_SERVICE\_PORT\_PROJECT\_GRPC = 10010

PROJECT\_PORT = tcp://172.20.12.36:1234

PROJECT\_PORT\_10010\_TCP = tcp://172.20.12.36:10010

PROJECT\_PORT\_10010\_TCP\_ADDR = 172.20.12.36

PROJECT\_PORT\_10010\_TCP\_PORT = 10010

PROJECT\_PORT\_10010\_TCP\_PROTO = tcp

PROJECT\_PORT\_1234\_TCP = tcp://172.20.12.36:1234

PROJECT\_PORT\_1234\_TCP\_ADDR = 172.20.12.36

PROJECT\_PORT\_1234\_TCP\_PORT = 1234

PROJECT\_PORT\_1234\_TCP\_PROTO = tcp

PROJECT\_PORT\_80\_TCP = tcp://172.20.12.36:80

PROJECT\_PORT\_80\_TCP\_ADDR = 172.20.12.36

PROJECT\_PORT\_80\_TCP\_PORT = 80

PROJECT\_PORT\_80\_TCP\_PROTO = tcp

PROJECT\_PORT\_8080\_TCP = tcp://172.20.12.36:8080

PROJECT\_PORT\_8080\_TCP\_ADDR = 172.20.12.36

PROJECT\_PORT\_8080\_TCP\_PORT = 8080

PROJECT\_PORT\_8080\_TCP\_PROTO = tcp

PROJECT\_PORT\_9901\_TCP = tcp://172.20.12.36:9901

PROJECT\_PORT\_9901\_TCP\_ADDR = 172.20.12.36

PROJECT\_PORT\_9901\_TCP\_PORT = 9901

PROJECT\_PORT\_9901\_TCP\_PROTO = tcp

PROJECT\_SERVICE\_HOST = 172.20.12.36

PROJECT\_SERVICE\_PORT = 1234

PROJECT\_SERVICE\_PORT\_GRPC = 10010

PROJECT\_SERVICE\_PORT\_GRPC\_WEB = 8080

PROJECT\_SERVICE\_PORT\_GRPC\_WEB\_ADMIN = 9901

PROJECT\_SERVICE\_PORT\_HTTP = 80

PROJECT\_SERVICE\_PORT\_PROM\_PUBLISHER = 1234

Prometheus:Enabled = true

Prometheus:Port = 1234

REFLECT\_COMPONENT = project

REFLECT\_TESTRUNNER\_WAIT = 1

REFLECTPROJECT\_Prometheus:Enabled = true

REFLECTPROJECT\_Security:LicenseValidationStartDate = 2019-11-01T00:00:00Z

REFLECTPROJECT\_SyncServer:Cloud:SupportsDecimation = true

Security:ApiKeyOptions:ValidEntitlements:0 = REVIEW

Security:ApiKeyOptions:ValidEntitlements:1 = REFLECT

Security:ApiKeyOptions:ValidEntitlements:2 = DEVELOP

Security:DeviceSession:DeviceAccessTokenDuration = 01:00:00

Security:DeviceSession:DeviceCodeDuration = 00:05:00

Security:DeviceSession:DeviceRefreshTokenDuration = 730.00:00:00

Security:DeviceSession:VerificationUri = https://dashboard.reflect.unity3d.com/activate

Security:EntitlementMapping:AnnotationCreationEntitlements:0 = DEVELOP

Security:EntitlementMapping:AnnotationCreationEntitlements:1 = REVIEW

Security:EntitlementMapping:AnnotationCreationEntitlements:2 = REFLECT

Security:EntitlementMapping:AnnotationCreationEntitlements:3 = DEPLOY

Security:EntitlementMapping:Apps:CUSTOM\_APP:ValidEntitlements:0 = REVIEW

Security:EntitlementMapping:Apps:CUSTOM\_APP:ValidEntitlements:1 = DEVELOP

Security:EntitlementMapping:Apps:CUSTOM\_APP:ValidEntitlements:2 = REVIEW.COORDINATION

Security:EntitlementMapping:Apps:CUSTOM\_APP:ValidEntitlements:3 = DEPLOY

Security:EntitlementMapping:Apps:REFLECT\_PUBLISHER:LicenseRequests:0 = REVIEW

Security:EntitlementMapping:Apps:REFLECT\_PUBLISHER:LicenseRequests:1 = DEVELOP

Security:EntitlementMapping:Apps:REFLECT\_PUBLISHER:LicenseRequests:2 = REFLECT

Security:EntitlementMapping:Apps:REFLECT\_PUBLISHER:ValidEntitlements:0 = REVIEW

Security:EntitlementMapping:Apps:REFLECT\_PUBLISHER:ValidEntitlements:1 = DEVELOP

Security:EntitlementMapping:Apps:REFLECT\_REFERENCE\_VIEWER:ValidEntitlements:0 = REVIEW

Security:EntitlementMapping:Apps:REFLECT\_REFERENCE\_VIEWER:ValidEntitlements:1 = DEVELOP

Security:EntitlementMapping:Apps:REFLECT\_REFERENCE\_VIEWER:ValidEntitlements:2 = REVIEW.COORDINATION

Security:EntitlementMapping:Apps:REFLECT\_REVIEW:ApiKey = apikeyplaceholder

Security:EntitlementMapping:Apps:REFLECT\_REVIEW:ValidEntitlements:0 = REVIEW

Security:EntitlementMapping:Apps:REFLECT\_REVIEW:ValidEntitlements:1 = DEVELOP

Security:EntitlementMapping:Apps:REFLECT\_REVIEW:ValidEntitlements:2 = REVIEW.COORDINATION

Security:EntitlementMapping:Apps:REFLECT\_VIEWER:ValidEntitlements:0 = VIEWER\_LEGACY

Security:EntitlementMapping:Apps:UNITY\_EDITOR:ValidEntitlements:0 = DEVELOP

Security:EntitlementMapping:AutoAssigned:0 = VIEWER\_LEGACY

Security:EntitlementMapping:EntitlementPriority:DEVELOP = 0

Security:EntitlementMapping:EntitlementPriority:REVIEW = 0

Security:EntitlementMapping:EntitlementPriority:REVIEW.COORDINATION = 1

Security:EntitlementMapping:ManageProjectEntitlements:0 = DEVELOP

Security:EntitlementMapping:ManageProjectEntitlements:1 = REVIEW

Security:EntitlementMapping:ManageProjectEntitlements:2 = REFLECT

Security:EntitlementMapping:ManageProjectEntitlements:3 = DEPLOY

Security:EntitlementMapping:PresenterEntitlements:0 = DEVELOP

Security:EntitlementMapping:PresenterEntitlements:1 = REVIEW

Security:EntitlementMapping:PresenterEntitlements:2 = REFLECT

Security:EntitlementMapping:PresenterEntitlements:3 = DEPLOY

Security:EntitlementMapping:PublishEntitlements:0 = DEVELOP

Security:EntitlementMapping:PublishEntitlements:1 = REVIEW

Security:EntitlementMapping:PublishEntitlements:2 = REFLECT

Security:EntitlementMapping:PublishEntitlements:3 = DEPLOY

Security:EntitlementMapping:Remaps:REFLECT:0 = DEVELOP

Security:EntitlementMapping:Remaps:REFLECT:1 = REVIEW

Security:JEDunn:PublicKeyB64 =

Security:JEDunn:PublisherUserId =

Security:LicenseValidationStartDate = 2019-11-01T00:00:00Z

Security:OnPrem:AuthorizeEndpoint = http://unity.keycloak.local:8081/realms/reflect/protocol/openid-connect/auth

Security:OnPrem:ClientId = reflect

Security:OnPrem:ClientIdClaim = azp

Security:OnPrem:Domain = BAH OPR

Security:OnPrem:NameClaim = preferred\_username

Security:OnPrem:OAuthJwkUrl = http://unity.keycloak.local:8081/realms/reflect/protocol/openid-connect/certs

Security:OnPrem:OrganizationName = Tyndall Air Force Base

Security:OnPrem:RealmAccessRoleListOwnerOrManagerRole = ReflectOwnerManagerRole

Security:OnPrem:RealmAccessRoleListUserRole = ReflectUserRole

Security:OnPrem:RefreshEndpoint = http://unity.keycloak.local:8081/realms/reflect/protocol/openid-connect/token

Security:OnPrem:RevokeEndpoint = http://unity.keycloak.local:8081/realms/reflect/protocol/openid-connect/revoke

Security:OnPrem:Scope = openid

Security:OnPrem:TokenEndpoint = http://unity.keycloak.local:8081/realms/reflect/protocol/openid-connect/token

Security:OnPrem:UserInfoEndpoint = http://unity.keycloak.local:8081/realms/reflect/protocol/openid-connect/userinfo

Security:OnPrem:ValidIssuer = http://unity.keycloak.local:8081/realms/reflect

Security:OnPrem:ViewerProtocol = reflect

Security:SeatAllocationOptions:CleanupInterval = 24:00:00

* Recommendations:

It seems like your EKS (Elastic Kubernetes Service) deployment is revealing a lot of environment variables, including database connection strings, service host addresses, and security configurations related to Unity Reflect. Based on the details provided, there are a few possible points of attention:

**Key areas to investigate:**

1. **Sensitive Data Exposure**: The deployment logs are showing potentially sensitive information such as database credentials, API keys, and certificate details (development-certificate.pfx). You should ensure that sensitive data like this is being managed through Kubernetes secrets and not directly in environment variables or configuration files.

**Action**: Move sensitive data (e.g., connection strings, certificates) to Kubernetes Secrets or external secret management services such as AWS Secrets Manager or HashiCorp Vault.

1. **Service Discovery and Networking**: The log is exposing a lot of internal service addresses (e.g., 172.20.x.x). This implies service discovery issues. The services (e.g., RabbitMQ, PostgreSQL, Reflect components) seem to be correctly exposed via service ports.

**Action**: Verify that the services are properly defined in Kubernetes using Service and Ingress resources. Also, ensure that any networking policies, DNS, and load balancer configurations are functioning as expected.

1. **Environment Configuration**: The environment is currently set to Development. If this is a production environment, this setting should be changed to Production to ensure that proper logging, optimizations, and security measures are applied.

**Action**: Check your deployment YAML or Helm chart and modify the ASPNETCORE\_ENVIRONMENT or equivalent environment variable to Production if appropriate.

1. **Prometheus and OpenTelemetry**: It seems you have Prometheus enabled for monitoring (Prometheus:Enabled = true). If you are not seeing your service metrics in your monitoring setup, verify your service annotations, Prometheus scrape configs, and OpenTelemetry configurations.

**Action**: Ensure that Prometheus is scraping the correct service ports and that OpenTelemetry is enabled in your application code for appropriate telemetry data.

**Next Steps for Debugging:**

1. **Check Kubernetes Secrets**: If you’re not using secrets for sensitive data, create secrets using the command:

bash

Copy code

kubectl create secret generic db-credentials --from-literal=username=project\_service --from-literal=password=reflect

Then mount this secret in your deployment.

1. **Review Deployment Configurations**: Inspect your deployment YAML or Helm values file to check for misconfigurations:

yaml

Copy code

env:

- name: ASPNETCORE\_ENVIRONMENT

value: "Production"

1. **Networking Verification**: Use kubectl get svc and kubectl get pods to ensure that services are running properly and accessible.
2. **Logging and Debugging**: Review the logs of individual pods:

bash

Copy code

kubectl logs <pod-name>

Do you want help reviewing your Kubernetes configurations or a specific part of your deployment (e.g., service or ingress)?

* CODE

Security:SeatAllocationOptions:CollabLicenseEntitlement = REVIEW.COORDINATION

Security:SeatAllocationOptions:CollabSeatKeepAliveInterval = 00:00:10

Security:SeatAllocationOptions:CollabSeatTimeout = 00:00:20

Security:SeatAllocationOptions:FullLicenseEntitlement = REVIEW

Security:SeatAllocationOptions:ReviewSeatKeepAliveInterval = 00:00:10

Security:SeatAllocationOptions:ReviewSeatTimeout = 00:00:20

Security:TokenManager:Certificates:Folder = Server/Security/Authentication

Security:TokenManager:Certificates:Password = password1234

Security:TokenManager:NbfClaimOffset = 00:01:00

Security:UnityHubAccessTokenValidity = 30.00:00:00

Sentry:Dsn =

Sentry:IsEnvironmentUser = True

Sentry:SendDefaultPii = True

Serilog:MinimumLevel:Default = Debug

Serilog:MinimumLevel:Override:Microsoft.AspNetCore = Warning

Serilog:MinimumLevel:Override:Microsoft.EntityFrameworkCore = Warning

Serilog:MinimumLevel:Override:System.Net.Http.HttpClient = Warning

Serilog:WriteTo:0:Args:formatter = Serilog.Formatting.Compact.CompactJsonFormatter, Serilog.Formatting.Compact

Serilog:WriteTo:0:Args:serverUrl = http://localhost:5341

Serilog:WriteTo:0:Name = Seq

Serilog:WriteTo:1:Args:outputTemplate = [{Timestamp:HH:mm:ss} {Level:u3}] {Message:lj} <s:{SourceContext}>{NewLine}{Exception}

Serilog:WriteTo:1:Args:theme = Serilog.Sinks.SystemConsole.Themes.AnsiConsoleTheme::Code, Serilog.Sinks.Console

Serilog:WriteTo:1:Name = Console

ServerEndpoint:Host = 0.0.0.0

ServerEndpoint:Port = 10010

ServiceEndpoints:Annotations:HttpAddress = http://localhost:8000

ServiceEndpoints:MatchMaker:GrpcAddress = http://localhost:10042

ServiceEndpoints:MatchMaker:HttpAddress = http://localhost:5775

ServiceEndpoints:ProjectServer:GrpcAddress = http://localhost:10010

ServiceEndpoints:ProjectServer:GrpcWebAddress = http://localhost:8080

ServiceEndpoints:ProjectServer:HttpAddress = http://localhost:5555

ServiceEndpoints:VoipServer:OtherAddress = mumble://localhost:40000

Swagger:Enabled = True

SYNC\_API\_PORT = tcp://172.20.144.132:1234

SYNC\_API\_PORT\_10010\_TCP = tcp://172.20.144.132:10010

SYNC\_API\_PORT\_10010\_TCP\_ADDR = 172.20.144.132

SYNC\_API\_PORT\_10010\_TCP\_PORT = 10010

SYNC\_API\_PORT\_10010\_TCP\_PROTO = tcp

SYNC\_API\_PORT\_1234\_TCP = tcp://172.20.144.132:1234

SYNC\_API\_PORT\_1234\_TCP\_ADDR = 172.20.144.132

SYNC\_API\_PORT\_1234\_TCP\_PORT = 1234

SYNC\_API\_PORT\_1234\_TCP\_PROTO = tcp

SYNC\_API\_PORT\_80\_TCP = tcp://172.20.144.132:80

SYNC\_API\_PORT\_80\_TCP\_ADDR = 172.20.144.132

SYNC\_API\_PORT\_80\_TCP\_PORT = 80

SYNC\_API\_PORT\_80\_TCP\_PROTO = tcp

SYNC\_API\_SERVICE\_HOST = 172.20.144.132

SYNC\_API\_SERVICE\_PORT = 1234

SYNC\_API\_SERVICE\_PORT\_GRPC = 10010

SYNC\_API\_SERVICE\_PORT\_HTTP = 80

SYNC\_API\_SERVICE\_PORT\_PROM\_PUBLISHER = 1234

SYNC\_DECIMATOR\_PORT = tcp://172.20.23.33:1234

SYNC\_DECIMATOR\_PORT\_1234\_TCP = tcp://172.20.23.33:1234

SYNC\_DECIMATOR\_PORT\_1234\_TCP\_ADDR = 172.20.23.33

SYNC\_DECIMATOR\_PORT\_1234\_TCP\_PORT = 1234

SYNC\_DECIMATOR\_PORT\_1234\_TCP\_PROTO = tcp

SYNC\_DECIMATOR\_PORT\_80\_TCP = tcp://172.20.23.33:80

SYNC\_DECIMATOR\_PORT\_80\_TCP\_ADDR = 172.20.23.33

SYNC\_DECIMATOR\_PORT\_80\_TCP\_PORT = 80

SYNC\_DECIMATOR\_PORT\_80\_TCP\_PROTO = tcp

SYNC\_DECIMATOR\_SERVICE\_HOST = 172.20.23.33

SYNC\_DECIMATOR\_SERVICE\_PORT = 1234

SYNC\_DECIMATOR\_SERVICE\_PORT\_HTTP = 80

SYNC\_DECIMATOR\_SERVICE\_PORT\_PROM\_PUBLISHER = 1234

SYNC\_HLODBUILDER\_PORT = tcp://172.20.26.20:1234

SYNC\_HLODBUILDER\_PORT\_1234\_TCP = tcp://172.20.26.20:1234

SYNC\_HLODBUILDER\_PORT\_1234\_TCP\_ADDR = 172.20.26.20

SYNC\_HLODBUILDER\_PORT\_1234\_TCP\_PORT = 1234

SYNC\_HLODBUILDER\_PORT\_1234\_TCP\_PROTO = tcp

SYNC\_HLODBUILDER\_PORT\_80\_TCP = tcp://172.20.26.20:80

SYNC\_HLODBUILDER\_PORT\_80\_TCP\_ADDR = 172.20.26.20

SYNC\_HLODBUILDER\_PORT\_80\_TCP\_PORT = 80

SYNC\_HLODBUILDER\_PORT\_80\_TCP\_PROTO = tcp

SYNC\_HLODBUILDER\_SERVICE\_HOST = 172.20.26.20

SYNC\_HLODBUILDER\_SERVICE\_PORT = 1234

SYNC\_HLODBUILDER\_SERVICE\_PORT\_HTTP = 80

SYNC\_HLODBUILDER\_SERVICE\_PORT\_PROM\_PUBLISHER = 1234

SYNC\_MODELPROCESSOR\_PORT = tcp://172.20.139.182:1234

SYNC\_MODELPROCESSOR\_PORT\_1234\_TCP = tcp://172.20.139.182:1234

SYNC\_MODELPROCESSOR\_PORT\_1234\_TCP\_ADDR = 172.20.139.182

SYNC\_MODELPROCESSOR\_PORT\_1234\_TCP\_PORT = 1234

SYNC\_MODELPROCESSOR\_PORT\_1234\_TCP\_PROTO = tcp

SYNC\_MODELPROCESSOR\_PORT\_80\_TCP = tcp://172.20.139.182:80

SYNC\_MODELPROCESSOR\_PORT\_80\_TCP\_ADDR = 172.20.139.182

SYNC\_MODELPROCESSOR\_PORT\_80\_TCP\_PORT = 80

SYNC\_MODELPROCESSOR\_PORT\_80\_TCP\_PROTO = tcp

SYNC\_MODELPROCESSOR\_SAFEMODE\_PORT = tcp://172.20.106.34:1234

SYNC\_MODELPROCESSOR\_SAFEMODE\_PORT\_1234\_TCP = tcp://172.20.106.34:1234

SYNC\_MODELPROCESSOR\_SAFEMODE\_PORT\_1234\_TCP\_ADDR = 172.20.106.34

SYNC\_MODELPROCESSOR\_SAFEMODE\_PORT\_1234\_TCP\_PORT = 1234

SYNC\_MODELPROCESSOR\_SAFEMODE\_PORT\_1234\_TCP\_PROTO = tcp

SYNC\_MODELPROCESSOR\_SAFEMODE\_PORT\_80\_TCP = tcp://172.20.106.34:80

SYNC\_MODELPROCESSOR\_SAFEMODE\_PORT\_80\_TCP\_ADDR = 172.20.106.34

SYNC\_MODELPROCESSOR\_SAFEMODE\_PORT\_80\_TCP\_PORT = 80

SYNC\_MODELPROCESSOR\_SAFEMODE\_PORT\_80\_TCP\_PROTO = tcp

SYNC\_MODELPROCESSOR\_SAFEMODE\_SERVICE\_HOST = 172.20.106.34

SYNC\_MODELPROCESSOR\_SAFEMODE\_SERVICE\_PORT = 1234

SYNC\_MODELPROCESSOR\_SAFEMODE\_SERVICE\_PORT\_HTTP = 80

SYNC\_MODELPROCESSOR\_SAFEMODE\_SERVICE\_PORT\_PROM\_PUBLISHER = 1234

SYNC\_MODELPROCESSOR\_SERVICE\_HOST = 172.20.139.182

SYNC\_MODELPROCESSOR\_SERVICE\_PORT = 1234

SYNC\_MODELPROCESSOR\_SERVICE\_PORT\_HTTP = 80

SYNC\_MODELPROCESSOR\_SERVICE\_PORT\_PROM\_PUBLISHER = 1234

SYNC\_WORKER\_PORT = tcp://172.20.176.216:1234

SYNC\_WORKER\_PORT\_1234\_TCP = tcp://172.20.176.216:1234

SYNC\_WORKER\_PORT\_1234\_TCP\_ADDR = 172.20.176.216

SYNC\_WORKER\_PORT\_1234\_TCP\_PORT = 1234

SYNC\_WORKER\_PORT\_1234\_TCP\_PROTO = tcp

SYNC\_WORKER\_PORT\_80\_TCP = tcp://172.20.176.216:80

SYNC\_WORKER\_PORT\_80\_TCP\_ADDR = 172.20.176.216

SYNC\_WORKER\_PORT\_80\_TCP\_PORT = 80

SYNC\_WORKER\_PORT\_80\_TCP\_PROTO = tcp

SYNC\_WORKER\_SERVICE\_HOST = 172.20.176.216

SYNC\_WORKER\_SERVICE\_PORT = 1234

SYNC\_WORKER\_SERVICE\_PORT\_HTTP = 80

SYNC\_WORKER\_SERVICE\_PORT\_PROM\_PUBLISHER = 1234

SyncServer:Cloud:Address = http://localhost:7777

SyncServer:Cloud:ClientId = unity.sync-service.api.local

SyncServer:Cloud:ClientSecret = syncservice-secret-key

SyncServer:Cloud:SupportsDecimation = true

SyncServer:Cloud:SupportsHttp = True

SyncServer:HiddenAfter = 00:03:00

SyncServer:KeepAliveFrequency = 00:01:00

SyncServer:OfflineAfter = 00:02:00

SyncServer:RenewTokenAfter = 00:05:00

SyncServer:TokenDuration = 00:10:00

UnityApi:RetryPolicy:InitialDelay = 00:00:00.5

UnityApi:RetryPolicy:MaxRetries = 5

URLS = http://+:80

[19:24:56 DBG] Hosting starting <s:Microsoft.Extensions.Hosting.Internal.Host>

[19:24:56 WRN] No XML encryptor configured. Key {b46a9671-a43b-48e5-9426-9c00e110d7a5} may be persisted to storage in unencrypted form. <s:Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager>

[19:24:56 INF] Starting Prometheus Exporter on port 1234 <s:Unity.Opr.ProjectServer.Server.Grpc.ServerHost>

[19:24:56 INF] Grpc Server starting [0.0.0.0:10010] <s:Unity.Opr.ProjectServer.Server.Grpc.ServerHost>

[19:24:56 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>

* Notes:

The continuation of the code you're showing relates to the configuration of a Unity-based server application that involves several key elements related to security, logging, API services, and network configurations.

**Key Insights:**

1. **Security and Token Management**:
   * The configuration defines seat allocation for collaboration and review licenses, with timeouts and keep-alive intervals.
   * Certificates for token management are stored in a specific folder (Server/Security/Authentication) and are protected by a password (password1234). This could be sensitive information and should be handled carefully in production environments.
   * A specific NbfClaimOffset is set to prevent token abuse, and the Unity Hub access token has a long validity of 30 days.
2. **Logging (Serilog)**:
   * The application uses **Serilog** for structured logging.
   * The minimum log level is set to **Debug** by default but raised to **Warning** for certain components like Microsoft.AspNetCore and HttpClient.
   * Logging is directed to multiple sinks, including Seq (a server-based log collector) and the console, with custom output formatting.
   * Prometheus exporter is configured on port 1234 for monitoring, indicating integration with **Prometheus** to collect metrics.
3. **Service Endpoints**:
   * Multiple service endpoints are defined for internal communication (Annotations, MatchMaker, ProjectServer, VoipServer). These services communicate over both gRPC and HTTP on various local addresses and ports.
   * There are multiple SYNC-related ports and services with associated IP addresses, likely representing different components of a distributed system for synchronous operations.
4. **Cloud Sync Server**:
   * The **SyncServer** is configured with an address (http://localhost:7777) and client credentials (ClientId and ClientSecret). This likely indicates cloud-based functionality for synchronizing data across different services or clients.
   * The server supports decimation (likely data reduction for efficiency) and HTTP communication.
5. **API Retry Policies**:
   * Retry policies are configured for the Unity API with an initial delay of 0.5 seconds and a maximum of 5 retries. This helps in handling transient failures or network instability when communicating with the API.
6. **Warning and Debug Messages**:
   * A warning is raised regarding XML encryption, indicating that encryption is not configured properly for storing keys.
   * There's also a warning from **Entity Framework**, advising the removal of certain calls (AddEntityFramework\*) that may not be necessary and could cause conflicts.
7. **Server Startup**:
   * The server is starting on multiple endpoints (0.0.0.0:10010 for gRPC and +:80 for general HTTP). These are likely the main interfaces for external communication.
   * The project server has a gRPC service starting on 0.0.0.0:10010, indicating it’s listening for gRPC requests on this port.

**Overall:**

This configuration sets up a robust, scalable server that integrates multiple components like gRPC services, structured logging (via Serilog), secure token management, and a retry policy for API interactions. There are warnings to address related to security (encryption) and database context usage (Entity Framework), but otherwise, the system seems well-configured for a distributed, cloud-based environment.

* Issue

[19:24:57 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>

[19:24:57 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>

[19:24:57 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>

[19:24:57 ERR] An error occurred using the connection to database 'ProjectService' on server ''. <s:Microsoft.EntityFrameworkCore.Database.Connection>

[19:24:57 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 ---

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

at System.Threading.Tasks.Task.ExecuteWithThreadLocal(Task& currentTaskSlot, Thread threadPoolThread)

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

at Npgsql.TaskExtensions.ExecuteWithTimeout[TResult](Func`2 func, NpgsqlTimeout timeout, CancellationToken cancellationToken)

at Npgsql.TaskExtensions.WithCancellation[T](Task`1 task, CancellationToken cancellationToken)

at Npgsql.TaskExtensions.WithTimeout[T](Task`1 task, NpgsqlTimeout timeout)

at Npgsql.Internal.NpgsqlConnector.ConnectAsync(NpgsqlTimeout timeout, CancellationToken cancellationToken)

at Npgsql.Internal.NpgsqlConnector.RawOpen(SslMode sslMode, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken, Boolean isFirstAttempt)

at Npgsql.Internal.NpgsqlConnector.<Open>g\_\_OpenCore|191\_1(NpgsqlConnector conn, SslMode sslMode, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken, Boolean isFirstAttempt)

at Npgsql.Internal.NpgsqlConnector.Open(NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken)

at Npgsql.ConnectorPool.OpenNewConnector(NpgsqlConnection conn, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken)

at Npgsql.ConnectorPool.<Get>g\_\_RentAsync|28\_0(NpgsqlConnection conn, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken)

at Npgsql.NpgsqlConnection.<Open>g\_\_OpenAsync|45\_0(Boolean async, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenInternalAsync(Boolean errorsExpected, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenInternalAsync(Boolean errorsExpected, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenAsync(CancellationToken cancellationToken, Boolean errorsExpected)

at Microsoft.EntityFrameworkCore.Storage.RelationalCommand.ExecuteNonQueryAsync(RelationalCommandParameterObject parameterObject, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.RelationalDatabaseFacadeExtensions.ExecuteSqlRawAsync(DatabaseFacade databaseFacade, String sql, IEnumerable`1 parameters, CancellationToken cancellationToken)

at Unity.Opr.ProjectServer.API.Data.ProjectServiceDbContext.CleanUpSeatAllocations(CancellationToken cancelToken) in /app/Projects/ProjectServer/Core/Data/ProjectServiceDbContext.cs:line 251

at Unity.Opr.ProjectServer.API.Data.DataCleanupService.ExecuteAsync(CancellationToken cancelToken) in /app/Projects/ProjectServer/Core/Data/DataCleanupService.cs:line 51

[19:24:58 ERR] An error occurred using the connection to database 'ProjectService' on server ''. <s:Microsoft.EntityFrameworkCore.Database.Connection>

[19:24:58 ERR] An exception occurred while iterating over the results of a query for context type 'Unity.Opr.ProjectServer.API.Data.ProjectServiceDbContext'.

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 ---

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

at System.Threading.Tasks.Task.ExecuteWithThreadLocal(Task& currentTaskSlot, Thread threadPoolThread)

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

at Npgsql.TaskExtensions.ExecuteWithTimeout[TResult](Func`2 func, NpgsqlTimeout timeout, CancellationToken cancellationToken)

at Npgsql.TaskExtensions.WithCancellation[T](Task`1 task, CancellationToken cancellationToken)

at Npgsql.TaskExtensions.WithTimeout[T](Task`1 task, NpgsqlTimeout timeout)

at Npgsql.Internal.NpgsqlConnector.ConnectAsync(NpgsqlTimeout timeout, CancellationToken cancellationToken)

at Npgsql.Internal.NpgsqlConnector.RawOpen(SslMode sslMode, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken, Boolean isFirstAttempt)

at Npgsql.Internal.NpgsqlConnector.<Open>g\_\_OpenCore|191\_1(NpgsqlConnector conn, SslMode sslMode, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken, Boolean isFirstAttempt)

at Npgsql.Internal.NpgsqlConnector.Open(NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken)

at Npgsql.ConnectorPool.OpenNewConnector(NpgsqlConnection conn, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken)

at Npgsql.ConnectorPool.<Get>g\_\_RentAsync|28\_0(NpgsqlConnection conn, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken)

at Npgsql.NpgsqlConnection.<Open>g\_\_OpenAsync|45\_0(Boolean async, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenInternalAsync(Boolean errorsExpected, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenInternalAsync(Boolean errorsExpected, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenAsync(CancellationToken cancellationToken, Boolean errorsExpected)

at Microsoft.EntityFrameworkCore.Storage.RelationalCommand.ExecuteReaderAsync(RelationalCommandParameterObject parameterObject, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Query.Internal.SingleQueryingEnumerable`1.AsyncEnumerator.InitializeReaderAsync(AsyncEnumerator enumerator, CancellationToken cancellationToken)

at Npgsql.EntityFrameworkCore.PostgreSQL.Storage.Internal.NpgsqlExecutionStrategy.ExecuteAsync[TState,TResult](TState state, Func`4 operation, Func`4 verifySucceeded, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Query.Internal.SingleQueryingEnumerable`1.AsyncEnumerator.MoveNextAsync() <s:Microsoft.EntityFrameworkCore.Query>

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 ---

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

at System.Threading.Tasks.Task.ExecuteWithThreadLocal(Task& currentTaskSlot, Thread threadPoolThread)

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

at Npgsql.TaskExtensions.ExecuteWithTimeout[TResult](Func`2 func, NpgsqlTimeout timeout, CancellationToken cancellationToken)

at Npgsql.TaskExtensions.WithCancellation[T](Task`1 task, CancellationToken cancellationToken)

at Npgsql.TaskExtensions.WithTimeout[T](Task`1 task, NpgsqlTimeout timeout)

at Npgsql.Internal.NpgsqlConnector.ConnectAsync(NpgsqlTimeout timeout, CancellationToken cancellationToken)

at Npgsql.Internal.NpgsqlConnector.RawOpen(SslMode sslMode, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken, Boolean isFirstAttempt)

at Npgsql.Internal.NpgsqlConnector.<Open>g\_\_OpenCore|191\_1(NpgsqlConnector conn, SslMode sslMode, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken, Boolean isFirstAttempt)

at Npgsql.Internal.NpgsqlConnector.Open(NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken)

at Npgsql.ConnectorPool.OpenNewConnector(NpgsqlConnection conn, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken)

at Npgsql.ConnectorPool.<Get>g\_\_RentAsync|28\_0(NpgsqlConnection conn, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken)

at Npgsql.NpgsqlConnection.<Open>g\_\_OpenAsync|45\_0(Boolean async, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenInternalAsync(Boolean errorsExpected, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenInternalAsync(Boolean errorsExpected, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenAsync(CancellationToken cancellationToken, Boolean errorsExpected)

at Microsoft.EntityFrameworkCore.Storage.RelationalCommand.ExecuteReaderAsync(RelationalCommandParameterObject parameterObject, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Query.Internal.SingleQueryingEnumerable`1.AsyncEnumerator.InitializeReaderAsync(AsyncEnumerator enumerator, CancellationToken cancellationToken)

at Npgsql.EntityFrameworkCore.PostgreSQL.Storage.Internal.NpgsqlExecutionStrategy.ExecuteAsync[TState,TResult](TState state, Func`4 operation, Func`4 verifySucceeded, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Query.Internal.SingleQueryingEnumerable`1.AsyncEnumerator.MoveNextAsync()

Resource associated with Metric:

service.name: ProjectServer

service.instance.id: 94f14544-02d2-4c80-9d13-4e770da6b676

telemetry.sdk.name: opentelemetry

telemetry.sdk.language: dotnet

telemetry.sdk.version: 1.7.0

Unhandled exception. 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 ---

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

at System.Threading.Tasks.Task.ExecuteWithThreadLocal(Task& currentTaskSlot, Thread threadPoolThread)

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

at Npgsql.TaskExtensions.ExecuteWithTimeout[TResult](Func`2 func, NpgsqlTimeout timeout, CancellationToken cancellationToken)

at Npgsql.TaskExtensions.WithCancellation[T](Task`1 task, CancellationToken cancellationToken)

at Npgsql.TaskExtensions.WithTimeout[T](Task`1 task, NpgsqlTimeout timeout)

at Npgsql.Internal.NpgsqlConnector.ConnectAsync(NpgsqlTimeout timeout, CancellationToken cancellationToken)

at Npgsql.Internal.NpgsqlConnector.RawOpen(SslMode sslMode, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken, Boolean isFirstAttempt)

at Npgsql.Internal.NpgsqlConnector.<Open>g\_\_OpenCore|191\_1(NpgsqlConnector conn, SslMode sslMode, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken, Boolean isFirstAttempt)

at Npgsql.Internal.NpgsqlConnector.Open(NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken)

at Npgsql.ConnectorPool.OpenNewConnector(NpgsqlConnection conn, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken)

at Npgsql.ConnectorPool.<Get>g\_\_RentAsync|28\_0(NpgsqlConnection conn, NpgsqlTimeout timeout, Boolean async, CancellationToken cancellationToken)

at Npgsql.NpgsqlConnection.<Open>g\_\_OpenAsync|45\_0(Boolean async, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenInternalAsync(Boolean errorsExpected, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenInternalAsync(Boolean errorsExpected, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Storage.RelationalConnection.OpenAsync(CancellationToken cancellationToken, Boolean errorsExpected)

at Microsoft.EntityFrameworkCore.Storage.RelationalCommand.ExecuteReaderAsync(RelationalCommandParameterObject parameterObject, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Query.Internal.SingleQueryingEnumerable`1.AsyncEnumerator.InitializeReaderAsync(AsyncEnumerator enumerator, CancellationToken cancellationToken)

at Npgsql.EntityFrameworkCore.PostgreSQL.Storage.Internal.NpgsqlExecutionStrategy.ExecuteAsync[TState,TResult](TState state, Func`4 operation, Func`4 verifySucceeded, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Query.Internal.SingleQueryingEnumerable`1.AsyncEnumerator.MoveNextAsync()

at Microsoft.EntityFrameworkCore.Query.ShapedQueryCompilingExpressionVisitor.SingleOrDefaultAsync[TSource](IAsyncEnumerable`1 asyncEnumerable, CancellationToken cancellationToken)

at Microsoft.EntityFrameworkCore.Query.ShapedQueryCompilingExpressionVisitor.SingleOrDefaultAsync[TSource](IAsyncEnumerable`1 asyncEnumerable, CancellationToken cancellationToken)

at Unity.Opr.ProjectServer.ExternalServices.OnPrem.OnPremOrganizationAndOAuth2ConfigurationService.StartAsync(CancellationToken cancellationToken) in /app/Projects/ProjectServer/ExternalServices/OnPrem/OnPremOrganizationAndOAuth2ConfigurationService.cs:line 41

at Microsoft.Extensions.Hosting.Internal.Host.StartAsync(CancellationToken cancellationToken)

at Microsoft.Extensions.Hosting.HostingAbstractionsHostExtensions.RunAsync(IHost host, CancellationToken token)

at Microsoft.Extensions.Hosting.HostingAbstractionsHostExtensions.RunAsync(IHost host, CancellationToken token)

at Microsoft.Extensions.Hosting.HostingAbstractionsHostExtensions.Run(IHost host)

at Unity.Opr.ProjectServer.Program.Main(String[] args) in /app/Projects/ProjectServer/Program.cs:line 32 can you narrow what these logs mean and how do I resolve this?

* Soln

It looks like you're encountering a series of issues related to database connections and Entity Framework Core warnings and errors. Here’s a breakdown of the key problems in the logs:

**Key Issues:**

1. **Sensitive Data Logging Enabled:**
   * [19:24:57 WRN] Sensitive data logging is enabled.
   * This warning indicates that sensitive data logging is active, which should only be used in development environments. Make sure to disable this in production to avoid exposing sensitive information.
2. **Foreign Key Mapping Conflict:**
   * [19:24:57 WRN] The foreign key property 'DeployIdPViewerBundleId.DeployIdPId1' was created in shadow state...
   * This suggests a conflict with a foreign key mapping in your Entity Framework model. You may need to review your DeployIdPViewerBundleId entity mapping and adjust the DeployIdPId property to ensure it’s mapped correctly.
3. **Default Value for Non-Nullable Bool:**
   * [19:24:57 WRN] The 'bool' property 'IsOnline' on entity type 'SyncServer' is configured with a database-generated default.
   * The IsOnline property is not nullable (bool instead of bool?), and the default value for false will always be applied. If you want to allow for a null state, change the type to bool?.
4. **Database Connection Errors:**
   * [19:24:57 ERR] An error occurred using the connection to database 'ProjectService' on server ''.
   * There is no server name specified, which may point to a configuration issue. Check your database connection string in your application settings to ensure it’s properly configured.
5. **DNS Resolution Error:**
   * [System.Net.Internals.SocketExceptionFactory+ExtendedSocketException]: Name or service not known
   * This is a DNS-related error. The system is unable to resolve the host name for the database server. This could be due to:
     + Incorrect DNS settings.
     + The database server not being reachable.
     + A misconfigured hostname in the connection string.
6. **Floating Seat Database Cleanup Failure:**
   * [19:24:57 FTL] Unexpected exception during floating seat database cleanup
   * This fatal error occurs when the floating seat cleanup process is trying to run. The root cause seems to be the database connection issue (System.Net.Internals.SocketExceptionFactory).

**Recommendations:**

1. **Fix the Database Connection String:**
   * Ensure that your connection string is correct and points to a valid database server. Look for potential errors in the configuration file that stores this string (e.g., appsettings.json or environment variables).
2. **Disable Sensitive Data Logging in Production:**
   * In your Startup.cs or Program.cs, make sure sensitive data logging is disabled when running in production mode:

csharp

Copy code

if (env.IsDevelopment())

{

options.EnableSensitiveDataLogging();

}

1. **Fix the Foreign Key Conflict:**
   * Review the foreign key mapping for DeployIdPViewerBundleId and ensure that the property DeployIdPId is correctly mapped and not conflicting with other properties.
2. **DNS Configuration Check:**
   * Verify that the database server hostname is correctly resolved. You may need to:
     + Check your DNS configuration.
     + Verify the database server's hostname in the connection string.
     + Ensure the database server is running and accessible from the application.
3. **Nullable bool? for Default Values:**
   * If the IsOnline field should use a database default, consider making it nullable (bool?). This way, the default value will only apply when the field is explicitly set to null.
4. **Review Floating Seat Cleanup Process:**
   * Since the error occurs during the floating seat cleanup, investigate the code handling this process. The cleanup method may be depending on the database connection that’s failing.

Let me know if you need specific guidance on any of these steps!

4o

### postgres::

**## install postgress**

**## update listening Ip**

- see image pg1

```

sudo vi  /var/lib/pgsql/data/postgressql.conf

  OR

sudo vi /etc/postgresql/16/main/postgresql.conf

```

- In that file, look for --> listen\_addresses = 'localhost'

- update to --> listen\_addresses = 'your-server-IP'

- In my case: update to --> listen\_addresses = '192/168.0.115'

**## update  Ipv4 local connection**

- see image pg2

- OR  see image pg2.1

```

sudo vi  /var/lib/pgsql/data/pg\_hba.conf

   OR

sudo vi /etc/postgresql/16/main/pg\_hba.conf

```

**## restart postgresql.service**

```

restart postgresql.service

ss -nlt | grep 5432

```

**## My existing Database for sonarqube**

```

## I use this to connect to database, NOT REQUIRED HERE:

psql -h cafanwi-postgres.sosotech.io -U cafanwiiuser -d sonarqube

```

**## Create a new database for jira using same user**

```

psql -h cafanwi-postgres.sosotech.io -U cafanwiiuser -d jira

   OR

psql -h 10.0.0.36 -U cafanwiiuser -d jira

```

---------------------------------

10.0.0.36

**## Getting started**

suso su - OR sudo su - postgres

psql

\l

**## Create password**

echo thisissosotech | base64

suso su - OR sudo su - postgres

psql

\l

sudo systemctl restart postgres

ALTER USER postgres PASSWORD 'dG9tYXRvZXNmcnVpdHNhbGFKCg';

**## Connect to PostgreSQL:**

psql -h <your\_postgres\_host> -U <your\_postgres\_user> -d cafanwii-db1

**## Create a Database:**

CREATE DATABASE <your\_database\_name>;

**## Create a User:**

CREATE USER <your\_user\_name> WITH ENCRYPTED PASSWORD '<your\_password>';

**## Grant Permissions:**

GRANT ALL PRIVILEGES ON DATABASE <your\_database\_name> TO <your\_user\_name>;

**#### Execute PostgreSQL commands**

\l            #  List databases

\c your\_database\_name   # Connect to a specific database

\dt           # List tables in the current database

SELECT current\_user;    # Show current user

SELECT current\_database();  # Show current database

SHOW search\_path;        # Show search path

\du           # List users

\q            # Exit the psql shell

-------

10.0.0.36

**## Getting started**

suso su - OR sudo su - postgres

psql

\l

**## Create password**

echo thisissosotech | base64

suso su - OR sudo su - postgres

psql

\l

sudo systemctl restart postgres

**## get host**

postgres=# SELECT current\_setting('listen\_addresses') AS host;

**## Get current database and user**

postgres=# SELECT current\_user AS user, current\_database() AS database;

**## Get hostname**

SELECT boot\_val,reset\_val

FROM pg\_settings

WHERE name='listen\_addresses';;

**## To change password**

ALTER USER postgres PASSWORD 'D20$';

**## connect to database, will require password**

psql -h localhost -U postgres -d postgres

psql -h localhost -U postgres -d cafanwii-db1

kubectl port-forward service/sonarqube 9000:9000

## chatted

Based on the provided PostgreSQL definitions, here are some potential issues and troubleshooting suggestions:

**Key Points to Consider**

1. **User Creation and Permissions**:
   * The script creates a user matchmaker\_service with a password stored in the environment variable $MATCHMAKER\_DB\_PASSWORD. Ensure that this variable is correctly set before running the script. If the password is empty or not set, the user creation will fail.
   * Make sure that the user executing this script has the necessary permissions to create users and databases.
2. **Database Creation**:
   * The script creates a database named Matchmaker. Ensure that the user matchmaker\_service has the necessary permissions on this database once it is created. If the permissions are insufficient, the application may fail to connect.
   * The database is created with template0, which is generally safe, but ensure that there are no other issues with the template.
3. **Error Handling**:
   * The use of set -e ensures that the script stops execution on any error. Check the logs to identify at which step the error occurs.
4. **Migrations and Table Creation**:
   * The migration script checks if the table already exists using IF NOT EXISTS. This is good practice, but if a table creation fails, the subsequent steps may not execute. Ensure the database is in a clean state for these migrations to work correctly.
5. **Transaction Management**:
   * The use of START TRANSACTION and COMMIT around critical sections helps in managing data integrity. However, if a transaction fails, it will not commit any changes. Monitor the transaction log for any issues.

### **Troubleshooting Commands**

psql -U <username> -h <hostname> -d <database\_name>

SELECT \* FROM pg\_stat\_activity;

\du

\l

SELECT d.datname AS "Database",

pg\_catalog.pg\_get\_userbyid(d.datdba) AS "Owner"

FROM pg\_catalog.pg\_database d;

### You can run the SQL script while enabling error detail logging:

psql -v ON\_ERROR\_STOP=1 -U <username> -f <script.sql>

######### **For Windows (Command Prompt or PowerShell)**

set POSTGRES\_USER=your\_postgres\_user

set MATCHMAKER\_DB\_PASSWORD=your\_password

$env:POSTGRES\_USER = "your\_postgres\_user"

$env:MATCHMAKER\_DB\_PASSWORD = "your\_password"

### **For Bash (Linux/Mac)**

export POSTGRES\_USER=your\_postgres\_user

export MATCHMAKER\_DB\_PASSWORD=your\_password