Report Portal

NOTE: this document is generated, do not edit manually. Source: https://github.com/viacomcbs/devops-docs

Introduction

According to documentation:

> *ReportPortal is a TestOps service, that provides increased capabilities to speed up results analysis and reporting through the use of built-in analytic features.*

For brevity “Report Portal” is abbreviated as “RP” in this documentation.

RP is deployed in 2 instances:

- production
- development

Architecture

RP installation consists of its own internal main components, described here and uses external ones for data persistence and processing.

External ones are:

- PostgreSQL database
- OpenSearch data search and analytics engine
- RabbitMQ message broker
- MinIO object storage

ArgoCD configuration

- Installation uses official RP Helm chart. It's named `reportportal` and is pulled from official RP Helm repo - https://reportportal.io/kubernetes/
- This chart uses external charts as its dependencies, i.e.:
  - `postgresql` from https://charts.bitnami.com/bitnami
  - `rabbitmq` from https://charts.bitnami.com/bitnami

    - `minio` from https://charts.bitnami.com/bitnami
    - `opensearch` from https://opensearch-project.github.io/helm-charts/
- K8s deployment is managed by ArgoCD.
- Both RP instances are installed in the `devops` ArgoCD tenant.
- All Helm chart repositories used directly and indirectly need to be added to allowed chart repositories in `devops` tenant. That is why there are following lines in the tenant's definition in devops-k8s/tenants/devops.yaml:

  ```
  allowedSourceRepos:
    # ... other repos not connected with RP
    - https://reportportal.io/kubernetes/
    - https://charts.bitnami.com/bitnami/postgresql/
    - https://charts.bitnami.com/bitnami/rabbitmq/
    - https://charts.bitnami.com/bitnami/minio/
    - https://opensearch-project.github.io/helm-charts/
    # ... other repos not connected with RP
  ```

- ArgoCD application definition devops-k8s-applications/applications/us-east-1/report-portal.yaml
- Core RP components are using images built only for x86 architecture. That is why there are so many `nodeSelector:` values passed to the main Helm chart, e.g. in devops-k8s-applications/report-portal/common.yaml:

  ```
  serviceapi:
  nodeSelector:
    paramount.tech/architecture: x86
  ```

Development instance

- Installed on the `use1-application-dev` cluster in `devops-dev` namespace.
- Uses official RP's Helm chart, v24.1.2
- Value files:
    - Common values for all instances
    - Values specific to development instance
- Installs all external components (i.e. PostgreSQL, OpenSearch, RabbitMQ and MinIO) as deployments on the same K8s cluster, using chart dependencies.
- All data stored on persistent volumes.

Production instance

- Installed on the `use1-application-prod` cluster in `devops-prod` namespace.

- Uses official RP's Helm chart, v24.1.7
- Value files:
  - Common values for all instances
  - Values specific to production instance
- Installs all external components (i.e. PostgreSQL, OpenSearch, RabbitMQ and MinIO) as AWS resources. Their Terraform definition is in devops-tf-aws/viacbs-networkstreaming-prod/us-east-1/devops/reportportal/prod.
  - Used AWS services are:
    - for PostgreSQL - Amazon RDS database (PostgreSQL)
    - for OpenSearch - Amazon OpenSearch Service domain
    - for RabbitMQ - Amazon MQ broker (RabbitMQ type)
    - for MinIO - Amazon S3 bucket
- There are no RP's persistent volumes on the K8s cluster.

Instalation on the K8s cluster

Development instance

1. Make sure that the certificate for reportportal.dev.tools.paramount.tech domain is created and is valid → devops-tf-aws/viacbs-networkstreaming-nonprod/us-east-1/devops/acm-certificates/dev/reportportal.dev.tools.paramount.tech.
2. Make sure that this domain and its certificate are registered on the `devops` tenant NLB → devops-k8s/tenants/devops.yaml in the section `allowedDestinations.clusterName["use1-application-dev"].internalIngress.nlb.config`.
3. Verify the Helm chart values in value files in devops-k8s-applications/report-portal.
4. Apply or synchronise the ArgoCD application in devops-k8s-applications/applications/us-east-1/report-portal.yaml for the cluster destination `clusterName: use1-application-dev`.

Production instance

1. Make sure that the certificate for reportportal.tools.paramount.tech domain is created and is valid → devops-tf-aws/viacbs-networkstreaming-prod/us-east-1/devops/acm-certificates/prod/reportportal.tools.paramount.tech.
2. Make sure that this domain and its certificate are registered on the `devops` tenant NLB → devops-k8s/tenants/devops.yaml in the section

`allowedDestinations.clusterName["use1-application-prod"].internalIngress.nlb.config`.

3. Check/create AWS resources in the following order:
   a. IAM user and roles - devops-tf-aws/viacbs-networkstreaming-prod/us-east-1/devops/reportportal/prod/irsa
   b. Secret store - devops-tf-aws/viacbs-networkstreaming-prod/us-east-1/devops/reportportal/prod/secret-store
   c. Verify the RDS database created by the DBA team - /devops-tf-aws/viacbs-networkstreaming-prod/us-east-1/devops/reportportal/prod/rds
   d. Locate The IAM user created in step 1) and verify whether it has access key and secret key created.
   e. Verify the secret store from step 2). It should contain following secrets:
      - POSTGRESQL_PASSWORD - it should contain the password for the user `reportportal_admin` created by the DBA team in the RDS instance from step 3), it should be taken from the `db/postgres/devops-reportportal-prod/app_secret` secret.
      - OPENSEARCH_PASSWORD - it should contain the password for the internal user on the OpenSearch domain.
      - OPENSEARCH_USER - it should contain the username for the internal user on the OpenSearch domain (i.e. `opensearch`).
      - RABBITMQ_PASSWORD - it should contain the password for the internal `rabbitmq` user on the RabbitMQ message broker.
      - ACCESS_KEY_ID - access key for the user from step 1).
      - SECRET_ACCESS_KEY - secret key for the user from step 1).
   f. Create / verify the rest of the resources from devops-tf-aws/viacbs-networkstreaming-prod/us-east-1/devops/reportportal/prod
   g. RabbitMQ broker should already contain its internal user and its password read from the RABBITMQ_PASSWORD secret.
   h. OpenSearch domain needs to have its internal user created from AWS console, as our OpenSearch AWS module does not serve this functionality yet. You can verify whether the user contained in the OPENSEARCH_USER and OPENSEARCH_PASSWORD secrets exists by interacting with the OpenSearch domain through curl, e.g.:

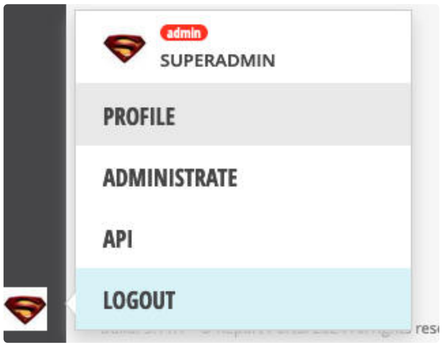
```
curl -u '<username>:<password>' https://vpc-devops-reportportal-prod-daetvnyo3aau2hyvmjhtqkzzqy.us-east-1.es.amazonaws.com/_cat/indices?v
```

4. Apply or synchronize ArgoCD application for K8s external secrets for ReportPortal - devops-k8s-applications/applications/us-east-1/report-portal-secret-store.yaml. You can view the mapping of the secret names between the AWS Secret Store and K8s secret in the value files:
   - devops-k8s-applications/report-portal-secret-store/common.yaml
   - devops-k8s-applications/report-portal-secret-store/use1-prod.yaml
5. Apply or synchronise the ArgoCD application in devops-k8s-applications/applications/us-east-1/report-portal.yaml for the cluster destination `clusterName: use1-application-prod`.

Okta integration

1. Out of the box there are 2 internal RP users: `superadmin` and `default`. First step should be logging into RP instance as those users and change their initial passwords. Initial password on the freshly installed RP instance for `default` is `1q2w3e` and intial password for `superadmin` is in Helm chart value `uat.superadminInitPasswd.password:`. Current passwords for RP are stored in Keeper in "Report Portal" folder.

   To change the password of a RP user you need to login as this user and go to its profile. Click on the user icon in the lower left corner of the portal and choose "PROFILE" and then "Change Password"

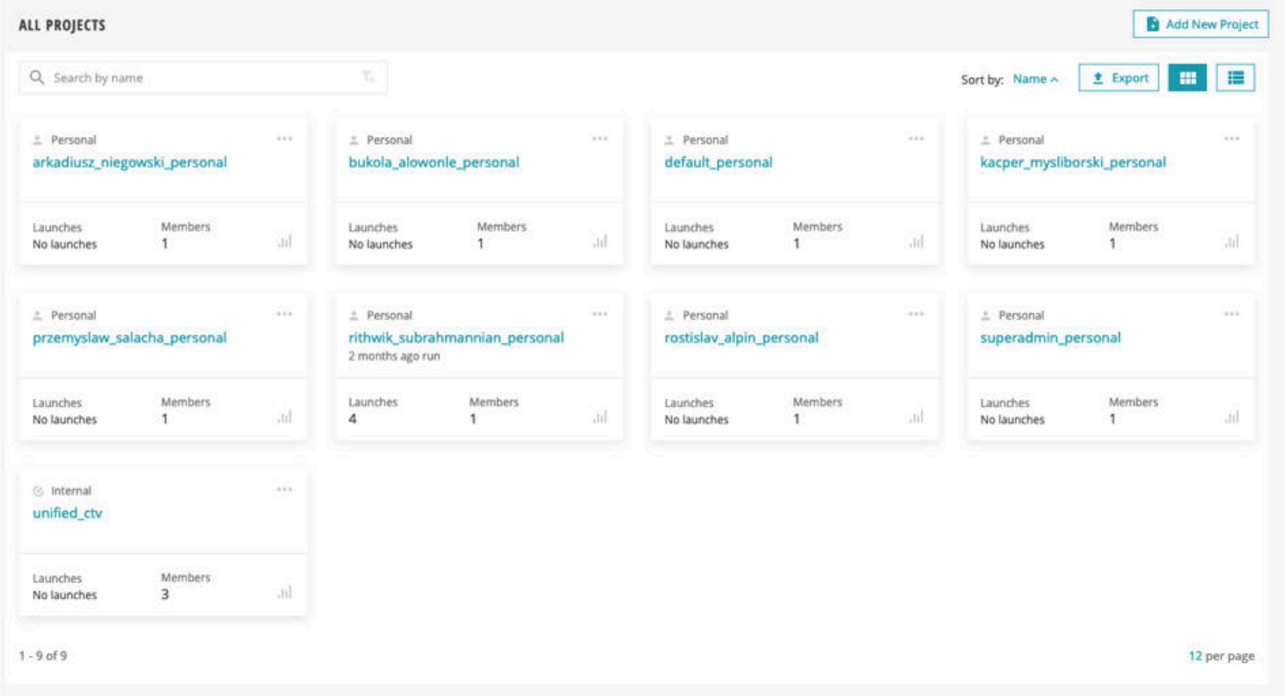
2. Request the creation of Okta applications for both RP instances through Pitstop ticket sent to OktaSupport@viacomcbs.com. Instructions for both Okta and RP itself are here. The Okta applications should be assigned to the AD groups "Okta-Paramount-ReportPortal-Users" (that's the group for regular users) and "Okta-Paramount-ReportPortal-Adm" (that's the group for the DevOps team). From the RP's point of view there is no functional difference between both those groups. The "Okta-Paramount-ReportPortal-Adm" exists only to be the AD management group for "Okta-Paramount-ReportPortal-Users" (it's in its "managed by" field).

   Currently Okta integration is set. There's no need to change it.

3. It's worth mentioning that membership in the group "Okta-Paramount-ReportPortal-Adm" doesn't automatically give you the admin rights in the RP. And only administrator is able to create shared non-personal projects and change other RP settings. Only the existing administrator (initially only `superadmin`) can grant admin right to another user. So if you log in to RP through Okta for the first time your RP user account gets created and you're not admin. You either need to ask another admin to "elevate" you or you should get `superadmin` password from Keeper and "elevate" yourself.

Project and user management

1. User management is done through AD group management and is decribed in RP [user documentation](#).
2. Project management can be done only as an admin user. You need to click on your user icon (similar to the password change process) and choose "ADMINISTRATE". You'll arrive on the projects management screen. Projects marked as "Personal" are individual users' projects. Shared projects are marked as "Internal"



Personal
arkadiusz_niegowski_personal
Launches No launches
Members 1

Personal
bukola_alowonle_personal
Launches No launches
Members 1

Personal
default_personal
Launches No launches
Members 1

Personal
kacper_mysliborski_personal
Launches No launches
Members 1

Personal
przemyslaw_salacha_personal
Launches No launches
Members 1

Personal
rithwik_subrahmannian_personal
2 months ago run
Launches 4
Members 1

Personal
rostislav_alpin_personal
Launches No launches
Members 1

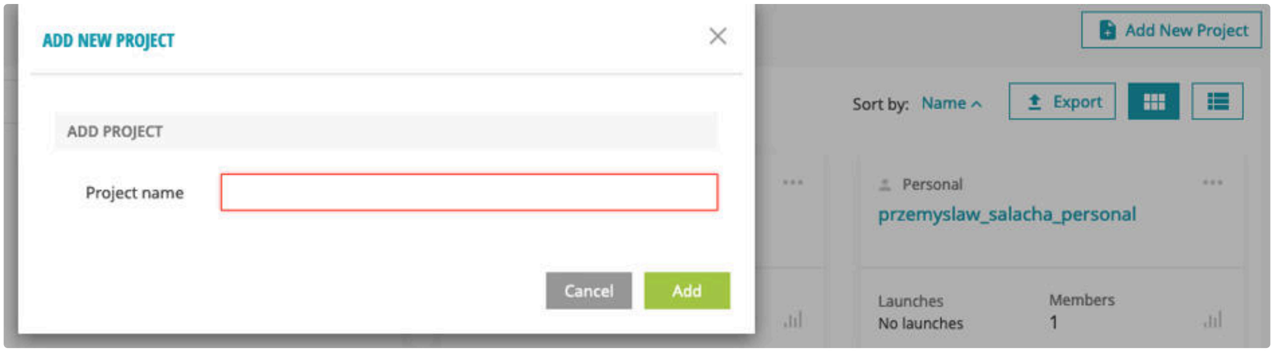
Personal
superadmin_personal
Launches No launches
Members 1

Internal
unified_ctv
Launches No launches
Members 3

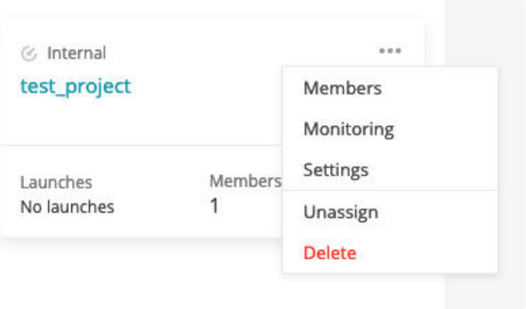
1 - 9 of 9

12 per page

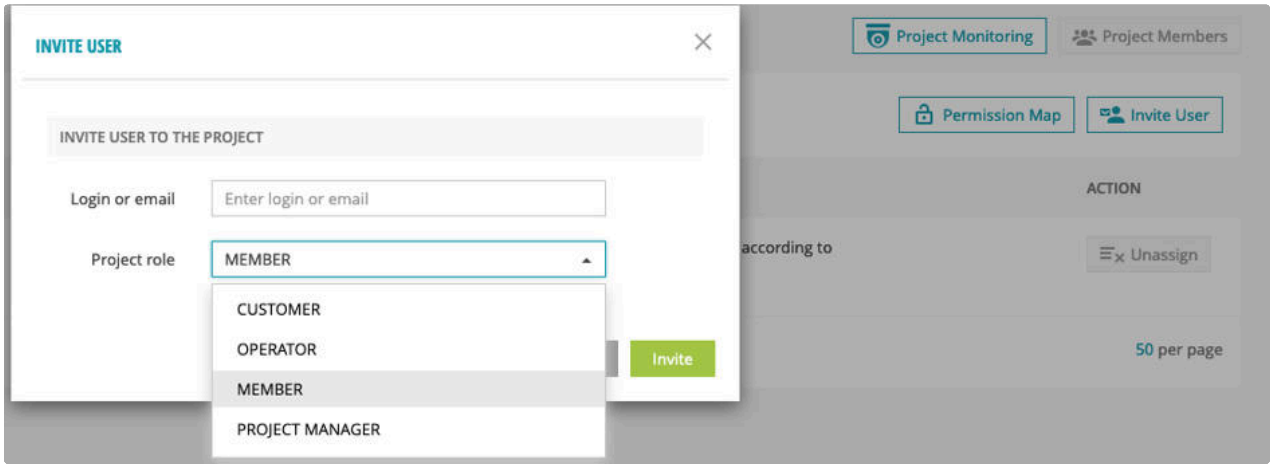
3. When you'll get a request to create project and assign some project managers to it, you should login to RP as administrator and go the "ADMINISTRATE" section.
   a. Then you should create a project.



b. Only after project's creation you can assign users to it. To do it go to the project's menu and select "Members".



c. You should then click "Invite user", select an user from the list and assign it to the chosen role and finally click on "Invite".



d. There are no actual invites sent through e-mail. User gets immediately assigned to the project. Users need to perform first time login to RP before they are visible in the list.