IAM components

The DOME IAM-Framework is a set of microservices, that enables users in the DOME ecosystem to authenticate into the DOME Marketplace. The authentication process itself is described further below in the Authentication section.

Overview and subcomponents

The DOME IAM-Framework consists of multiple open-source components. The components are not required to be used, as long as alternatives providing the same interfaces are used.

The IAM-Framework consists of following components:

Image not found or type unknown
IAM-components

- The Trusted Issuers List service provides an EBSI Trusted Issuers Registry implementation to act as the Trusted-List-Service in the DSBA Trust and IAM Framework. In addition, a Trusted-Issuers-List API is provided to manage the issuers.
- VCVerifier provides the necessary endpoints to offer SIOP-2/OIDC4VP compliant authentication flows. It exchanges VerifiableCredentials for JWT, that can be used for authorization and authentication in down-stream components.
- Credentials Config Service manages and provides information about services and the credentials they are using. It returns the scope to be requested from the wallet per service. Furthermore, it specifies the credentials required and the issuers list endpoints to validate against, when checking access for a certain service.
- The Keycloak-VC-Issuer is plugin for Keycloak to support SIOP-2/OIDC4VP clients and issue VerifiableCredentials through the OIDC4VCI-Protocol to compliant wallets.
- PDP is an implementation of a Policy-Decision Point, evaluating Json-Web-Tokens containing VerifiableCredentials in a DSBA-compliant way. It also supports the evaluation in the context of i4Trust.
- Keyrock is the FIWARE component responsible for Identity Management. Within DOME IAM-Framework, currently Keyrock is being used as the iSHARE-compliant Authorization Registry (see for details: https://dev.ishare.eu/delegation/endpoint.html), where attribute-based access policies are stored and used during the authorization process. Note, that this will be replaced by an ODRL-compliant policy registry. A description of the policies is given in the policies section.
- Kong Plugins allow to extend the API Gateway Kong by further functionalities. Kong Gateway is a lightweight, fast, and flexible cloud-native API gateway. One of the plugins is the PEP plugin, which is especially required within the IAM-components as PEP component and interacts with the PDP mentioned above.
- Waltid manages Keys, DIDs and VCs. It is used by VC Issuer and VCVerifier.

How to deploy

The recommended way of deployment is via the provided Helm charts.

To deploy a setup, the umbrella chart of the iam-components can be used as followed:

- create a configuration values file according to the own environment, as described here.

- add helm chart repository to helm installation

  helm repo add dome-iam https://dome-marketplace.github.io/iam-components
  helm repo update

  > ? All releases of the IAM-components reside in the helm-repository https://dome-marketplace.github.io/iam-components. In addition to that, all Pre-Release versions(build from the Pull Requests) are provided in the pre-repo https://dome-marketplace.github.io/iam-components/pre. The pre-repo will be cleaned-up from time to time, in order to keep the index manageable.

- install the components using the prepared configuration

  helm install <RELEASE_NAME> dome-iam/iam-components --namespace <NAME_SPACE> --version <CHART_VERSION> -f values.yaml

How to configure

The chart is released with a set of documented default values. The parameters listed below are important to set and should be updated at least:

- `rbac` and `serviceAccount`: Depending on your requirements, you might need to adapt settings for RBAC and service account
- `did`s of participants: Replace/add the DIDs of the issuer and other participants
- In the case of did:key provide correct key in keyfile.json for your issuer
- `keycloak.frontendUrl`: Externally accessible address of the keycloak (should be the same as defined in ingress/route)
- `keycloak.realm`: Adapt clients, users and roles according to your needs
- `<tir.com>`: replace everywhere with actual TIR URL
- `<dome-marketplace.org>`: replace with your own domain
- `keyrock.initData.scriptData`: Adapt the roles as in keycloak realm
- `kong.configMap`: Adapt the kong services and their routes

However, it is suggested to consult the respective charts listed below and check their documentation and configuration.

Component	Chart
postgresql	https://github.com/bitnami/charts/tree/main/bitnami/postgresql
mysql	https://github.com/bitnami/charts/tree/main/bitnami/mysql
vcwaltid	https://github.com/i4Trust/helm-charts/tree/main/charts/vcwaltid
keycloak	https://github.com/bitnami/charts/tree/main/bitnami/keycloak

Component	Chart
credentials-config-service	https://github.com/FIWARE/helm-charts/tree/main/charts/credentials-config-service
trusted-issuers-list	https://github.com/FIWARE/helm-charts/tree/main/charts/trusted-issuers-list
vcverifier	https://github.com/i4Trust/helm-charts/tree/main/charts/vcverifier
keyrock	https://github.com/FIWARE/helm-charts/tree/main/charts/keyrock
dsba-pdp	https://github.com/FIWARE/helm-charts/tree/main/charts/dsba-pdp
kong	https://github.com/Kong/charts/tree/main/charts/kong

How to validate a deployment

All components are configured with health and readiness checks to validate their own status, therefore being the base for a validation. These checks are utilized in the Kubernetes checks as defined in the helm charts.

How to operate

The underlying database service holds the persisted data and therefore requires a backup&recovery mechanism when operated in a production environment. The use of managed database is strongly encouraged for safety and convenience.

How to update

Upgrade to both a different chart version and new configuration can be accomplished with the following command

helm upgrade <RELEASE_NAME> dome-iam/iam-components --namespace <NAME_SPACE> --version <CHART_VERSION> -f values.yaml

Release process

Versioning of the main iam-components helm chart is handled based on the labels used in the pull requests used to introduce changes and is enforced in the build pipeline. The requester and reviewers must set the label according to the SemVer 2.0.0 versioning scheme.

Versioning of the components and sub-charts is recommended to use the same scheme.

Troubleshooting

> To be filled once feedback from integrators comes in

Revision #1
Created 23 May 2024 15:05:52
Updated 23 May 2024 15:06:53