# Siccar Microservices Deployment

# TLDR

It’s a bit all over the place as the services were written by different people with little forethought or coordination as to deployment (and other areas too… 😊). Only the Register service uses Mongo directly and so it does not use dapr.

# TODO Suggestions

* The tenant service should really use a datastore but because of licensing we’re stuck using a quite large json config file
* The wallet service seems to use a dapr secret but only for getting a key vault uri. This should be replaced by simple config env var

# Env Vars

|  |  |  |
| --- | --- | --- |
| Service | Var | Value |
| Action |  |  |
|  |  |  |
| Blueprint |  |  |
|  |  |  |
| Peer | PEER\_\_NAME |  |
|  | PEER\_SEED |  |
|  |  |  |
| Register | REGISTERREPOSITORY\_\_MONGODBSERVER | \* |
|  | REGISTERREPOSITORY\_\_DATABASENAME | siccar-mongo-dev |
|  |  |  |
| Tenant | ? |  |
|  |  |  |
| Validator | ? |  |
|  |  |  |
| Wallet |  |  |
|  |  |  |
|  |  |  |

\* mongodb://siccar-mongo-dev:Vjd5nBMxglQ2BFkAmpvWPPoBovPIiI1o8LB31FbCSgDR0UKu5Zu2OpFmnXiJWcruKQyfyIfhEbJ30G9UehWHkw==@siccar-mongo-dev.mongo.cosmos.azure.com:10255/?ssl=true&replicaSet=globaldb&retrywrites=false&maxIdleTimeMS=120000&appName=@siccar-mongo-dev@

I’ve left out log settings as they are a common across all the microservices.

# Dapr Component Files

|  |  |  |
| --- | --- | --- |
| Service | Dapr Yaml File | Name |
| Action | No data access configured to date | n/a |
|  |  |  |
| Blueprint | component-state-blueprint.yaml | blueprintstore |
|  |  |  |
| Peer | No data access configured to date | n/a |
|  |  |  |
| Register | Uses mongo directly, NOT dapr | n/a |
|  |  |  |
| Tenant | component-state-tenant.yaml | tenantstore |
|  |  |  |
| Validator | component-state-validator.yaml | validatorstore |
|  |  |  |
| Wallet | component-state-wallet.yaml | walletstore |
|  |  |  |
|  | component-pubsub.yaml | commonpubsub |
|  |  |  |

The convention is component followed by its dapr type and the service name if for a service. The register microservice does not use dapr for data access but has configuration for direct access to mongo db. The publish / subscribe component is common to all of the participating services to use.

These files are obviously related to the actual resources provisioned in Azure. Because we are choosing Azure as our cloud of choice, we will use a CosmosDB account, and each service will have its own collection.

# Microservice Deployment Files

|  |  |  |
| --- | --- | --- |
| Service | Yaml File | Name |
| Action | deployment-microservice-action.yaml |  |
|  |  |  |
| Blueprint | deployment-microservice-blueprint.yaml |  |
|  |  |  |
| Peer | deployment-microservice-peer.yaml |  |
|  |  |  |
| Register | deployment-microservice-register.yaml |  |
|  |  |  |
| Tenant | deployment-microservice-tenant.yaml |  |
|  |  |  |
| Validator | deployment-microservice-validator.yaml |  |
|  |  |  |
| Wallet | deployment-microservice-wallet.yaml |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Microservice Service Files

|  |  |  |
| --- | --- | --- |
| Service | Yaml File | Name |
| Action | service-microservice-action.yaml |  |
|  |  |  |
| Blueprint | service-microservice-blueprint.yaml |  |
|  |  |  |
| Peer | service-microservice-peer.yaml |  |
|  |  |  |
| Register | service-microservice-register.yaml |  |
|  |  |  |
| Tenant | service-microservice-tenant.yaml |  |
|  |  |  |
| Validator | service-microservice-validator.yaml |  |
|  |  |  |
| Wallet | service-microservice-wallet.yaml |  |
|  |  |  |
|  |  |  |
|  |  |  |