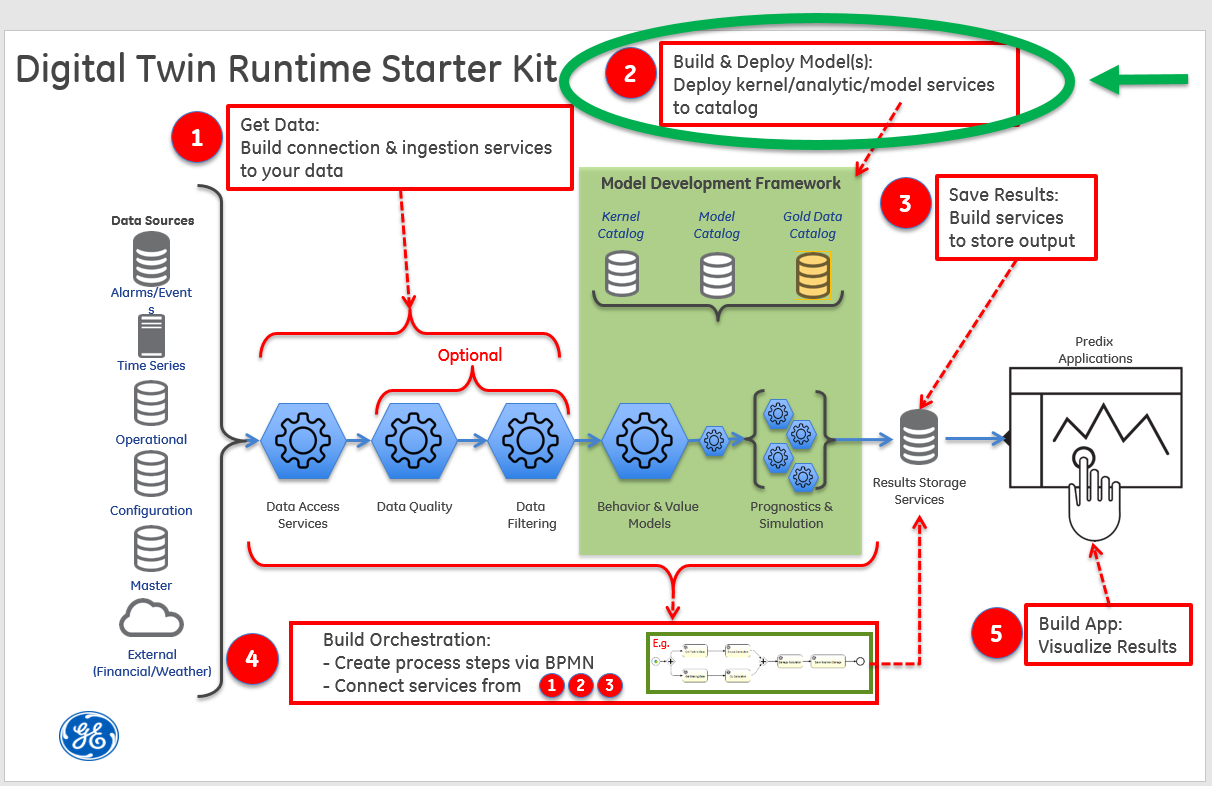
**Step 2: Build and Deploy Models**

**What you'll learn to do**



Step 2 focuses on how Digital Twin models and analytics are built, deployed, and run. Digital Twin models and analytics will be run via REST endpoints. In most cases, these models and analytics will be deployed to an instance of a[Predix Analytic Catalog](https://predix-io-dev.grc-apps.svc.ice.ge.com/services/service.html?id=1187)or an instance of a [Predix Analytic Runtime](https://predix-io-dev.grc-apps.svc.ice.ge.com/services/service.html?id=1174) and executed through a workflow orchestration (see [**Step 4:**](https://predix-io-dev.grc-apps.svc.ice.ge.com/resources/tutorials/tutorial-details.html?tutorial_id=1919&tag=1913&journey=Digital%20Twin%20Runtime%20Starter%20Kit&resources=1914,1915,1916,1917,1918,1919,1920)**Build Orchestration**). The Predix Analytics Catalog allows uploading models that are developed in Python, Java, or Matlab. For more information, refer to the Predix documentation on [Analytic Development Process](https://predix-io-dev.grc-apps.svc.ice.ge.com/docs/#Qd2kPYb7).

The simple workflow:



In this part of the tutorial, you will:

* Find the Predix-Zone-Id of the your Predix-Analytics-Catalog
* Create the **tutorial-actual-vs-expected-analytic application**
* Upload the analytic to the Analytic Catalog
* Execute the analytic hosted in the Analytic Catalog

**What you need to set up**

For this step, you will need an instance of the Analytics Catalog and an instance of the Predix UAA service. If you want to use the provided Postman collection to interact with the REST endpoints, you will need to set the authorization header with the bearer token for the UAA client that has permission to call the REST endpoint since the applications are secured. If you have not set up these services or secured the provided Postman collection, please see the **Getting Started** section for instructions.

**What you need to do**

**Find the Predix-Zone-Id of the your Predix-Analytics-Catalog**

In this part of the tutorial, you’ll need to know what your Predix-Zone-Id is for your Predix-Analytics-Catalog instance.

From the command line using the Cloud Foundry command line interface (cf cli) you’ll run commands to find the guid, which is the Predix Zone ID, of your instance of the predix-analytic-catalog service. For example:

|  |
| --- |
| D:\mydir\steam-turbine-tutorial\tutorial-svcs\tutorial-analytic> cf services Getting services in org DigitalTwin / space dev as 200000000@mail.ad.ge.com ... OK name                            service                             plan           bound apps           last operation ... my-analytics-catalog              predix-analytics-catalog   Bronze      tutorial-asset         create succeeded ... D:\mydir\steam-turbine-tutorial\tutorial-svcs\tutorial-analytic> cf service my-analytics-catalog --guid **69ab882f-ab6d-4c5a-8a4d-68f3a0148b22** D:\mydir\steam-turbine-tutorial\tutorial-svcs\tutorial-analytic> |

The guid string returned (for example, 69ab882f-ab6d-4c5a-8a4d-68f3a0148b22) is used as the value for the header parameter Predix-Zone-Id in Predix Analytic Catalog REST API calls. We’ll need this guid soon.

**Create the tutorial-analytic**

If you've already created the tutorial-asset application as described in **Step 1**: "Create the tutorial-asset to application," then you already have the code locally.

If you have not already built the code, you can do so with the following commands:

|  |
| --- |
| $ cd steam-turbine-tutorial/tutorial-analytics/tutorial-actual-vs-expected-analytic $ mvn clean package |

**Upload the analytic to the Analytic Catalog**

You can interact with the Predix Analytic Catalog through its REST API by using the provided Postman collection.

For all of the REST API calls, you will need to set the “Predix-Zone-Id” http header variable with the predix-zone-id, also known as the service GUID value, of your Analytic Catalog service instance. Here are the relevant API calls

|  |  |  |
| --- | --- | --- |
| **Step, Postman Request, and Sample URL** | **Header Params Required** | **Sample Body for POST Requests** |
| 1  POST Create catalog entry  https://predix-analytics-catalog-release.run.aws-usw02-pr.ice.predix.io/api/v1/catalog/analytics  The response body contains an “id”. This is the used as the catalogEntryId in the next step and as the <analytic-id> in steps 3 and 4. Note this analytic id as we’ll need it later when we want to execute the analytic. | |  |  | | --- | --- | | **Parameter** | **Value** | | Authorization | Bearer Token | | Predix-Zone-id | Analytic Catalog guid | | Content-type | application/json | | { "name":"steam-turbine-actual-vs-expected-analytic", "version":"v1", "supportedLanguage":"Java", "taxonomyLocation":"", "author":"Jane Doe", "description":"This analytic computes deltas between the actual (timeseries data) and expected (model predicted) temperatures values.", "customMetadata":"" } |
| 2    POST Upload analytic file/jar  https://predix-analytics-catalog-release.run.aws-usw02-pr.ice.predix.io/api/v1/catalog/artifacts | |  |  | | --- | --- | | **Parameter** | **Value** | | Authorization | Bearer Token | | Predix-Zone-id | Analytic Catalog guid | | |  |  |  | | --- | --- | --- | | **Form Field Name** | **Type** | **Value** | | file | file | Browse and select your file | | catalogEntryId | string / text | id from the created analytic entry (see above response) | | type | string / text | “executable” | | description. | string / text | describe the analytic | |
| 3  POST Validate your analytic  https://predix-analytics-catalog-release.run.aws-usw02-pr.ice.predix.io/api/v1/catalog/analytics/<analytic-id>/validation  The response body will contain a validationRequestId. This will be used in the next step as the <validation-request-id>. The analytic may take several minutes to validate. You can go ahead to the next step which will check the validation status of the analytic. | |  |  | | --- | --- | | **Parameter** | **Value** | | Authorization | Bearer Token | | Predix-Zone-id | Analytic Catalog guid | | Content-type | application/json | | { "series": [{ "timestamp": 1466308800000, "rpm": 3000, "actualTemperature": 201 }, { "timestamp": 1466308801000, "rpm": 3000, "actualTemperature": 203 }, { "timestamp": 1466308802000, "rpm": 3000, "actualTemperature": 208 }, { "timestamp": 1466308803000, "rpm": 3000, "actualTemperature": 211 }], "slope": 0.065, "intercept": 4.0 } |
| 4  GET Check the status of your analytic  https://predix-analytics-catalog-release.run.aws-usw02-pr.ice.predix.io/api/v1/catalog/analytics/<analytic-id>/deployment/<validation-request-id>  Note: The response body’s “status” field will change to “COMPLETED” when the validation has finished. | |  |  | | --- | --- | | **Parameter** | **Value** | | Authorization | Bearer Token | | Predix-Zone-id | Analytic Catalog guid | |  |

Note that Predix provides additional services for analytic support. These are the [Predix Analytics Runtime](https://www.predix.io/services/service.html?id=1174) and the [Predix Analytics User Interface](https://www.predix.io/services/service.html?id=1586). Predix has documented the [Analytic Development Process](https://www.predix.io/docs#Qd2kPYb7) which you can explore on your own.

**Execute the analytic hosted in the Analytic Catalog**

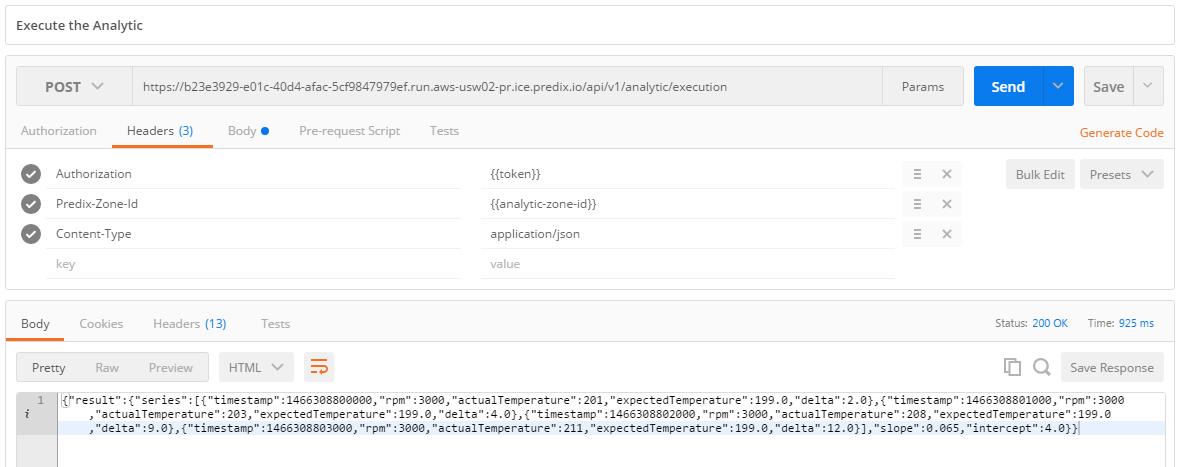
For the tutorial analytic, the POST request body for the analytic execution needs a JSON object consisting of an array (rpmSeries) of JSON objects consisting of a time stamp (timestamp), revolutions per minute (rpm), and actual temperature (actualTemperature) values followed by the slope and intercept values (coefficients for the regression model). An example of this JSON data is in the table below.

To execute your analytic, you will need the id of the deployed analytic. There’s a GET endpoint on the API in the [Analytics Catalog](https://www.predix.io/services/service.html?id=1187#/) called “/api/v1/catalog/analytics”. You can use this to retrieve the analytic id. You used this analytic id when you deployed your analytic via the REST API to the Analytic Catalog in the previous step.

You will also need a bearer token for your tutorial-svcs client and the Predix-Zone-Id for your Predix Analytics Catalog service. Both of which you should already have from completing earlier sections.

Now you can make the following POST request to the Analytic Catalog to execute your analytic:

|  |  |  |
| --- | --- | --- |
| **Postman Request, and Sample URL** | **Header Params Required** | **Sample Body for POST Requests** |
| POST Execute the analytic  https://<analytic-id>.run.aws-usw02-pr.ice.predix.io/api/v1/analytic/execution | |  |  | | --- | --- | | **Parameter** | **Value** | | Authorization | Bearer Token | | Predix-Zone-id | Analytic Catalog guid | | Content-type | application/json | | { "series": [{ "timestamp": 1466308800000, "rpm": 3000, "actualTemperature": 201 }, { "timestamp": 1466308801000, "rpm": 3000, "actualTemperature": 203 }, { "timestamp": 1466308802000, "rpm": 3000, "actualTemperature": 208 }, { "timestamp": 1466308803000, "rpm": 3000, "actualTemperature": 211 }], "slope": 0.065, "intercept": 4.0 } |



In **Step 4: Build Orchestration**, the BPMN file needs to know the above execution URL to run your analytic in the Digital Twin workflow.

**What you learned**

You learned how to find the predix zone id of your predix analytics catalog for use in the HTTP headers. You also learned how use the Predix Analytic Catalog to upload, validate, check the status of, and execute the pre-built analytic by using the catalog’s REST API.