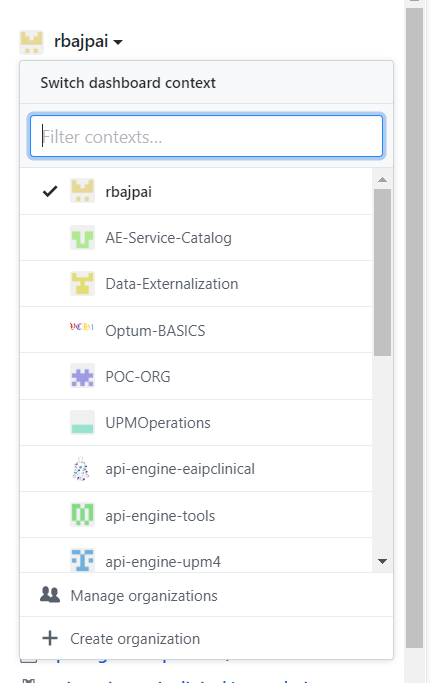
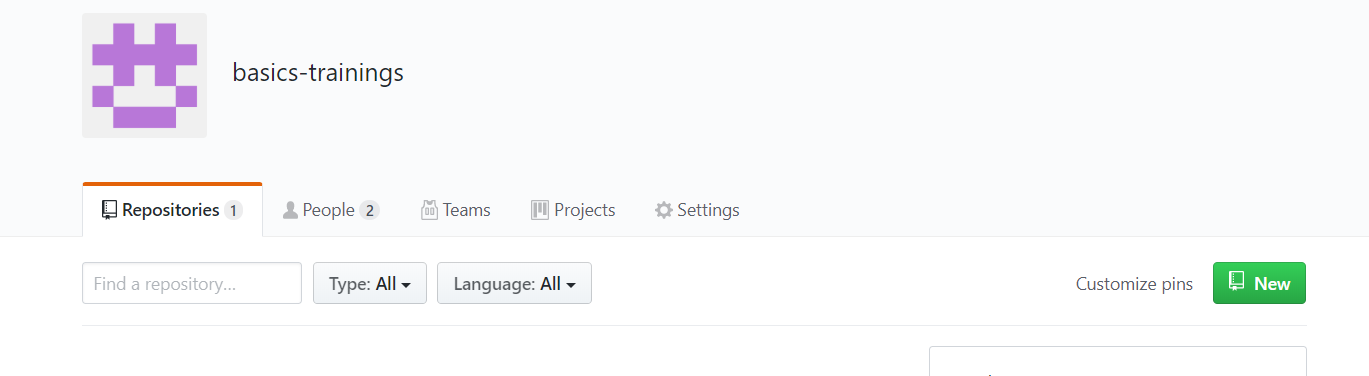
1. Create Github Organization.
2. Login to <https://github.optum.com/>
3. Click on create Organization

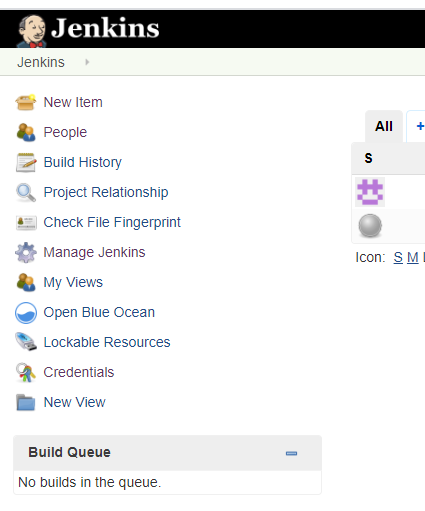


1. Create a repo on github organization.
2. Click on “new” or “create new repository”

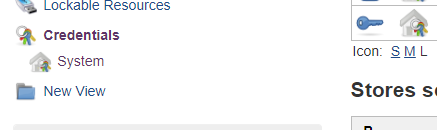


https://jenkins-basics.ocp-ctc-core-nonprod.optum.com/

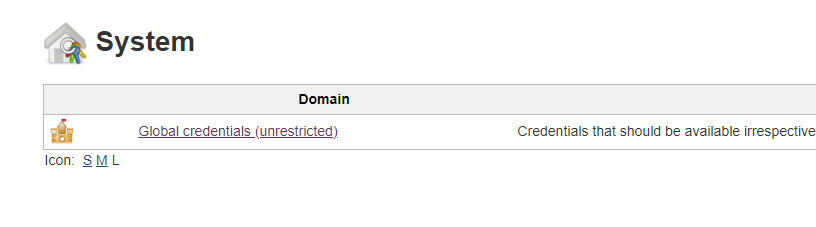
1. Add Credential’s in Jenkins – Add you MS ID
2. Click on credentials –



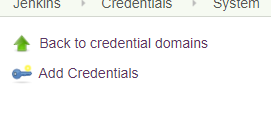
1. Click on system



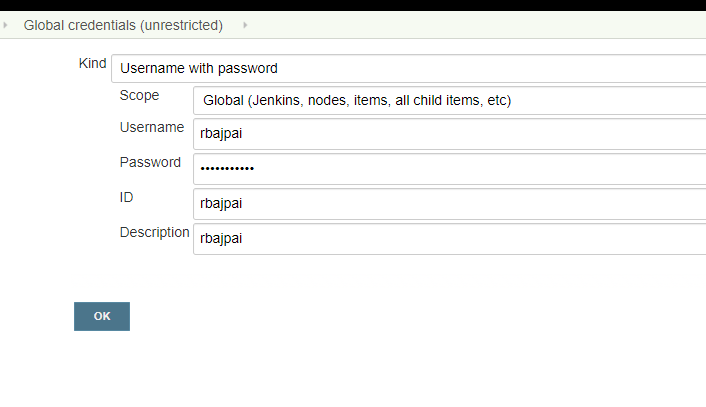
1. Click on Global credentials



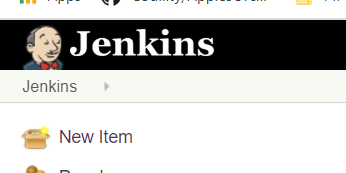
1. Click on Add credentials



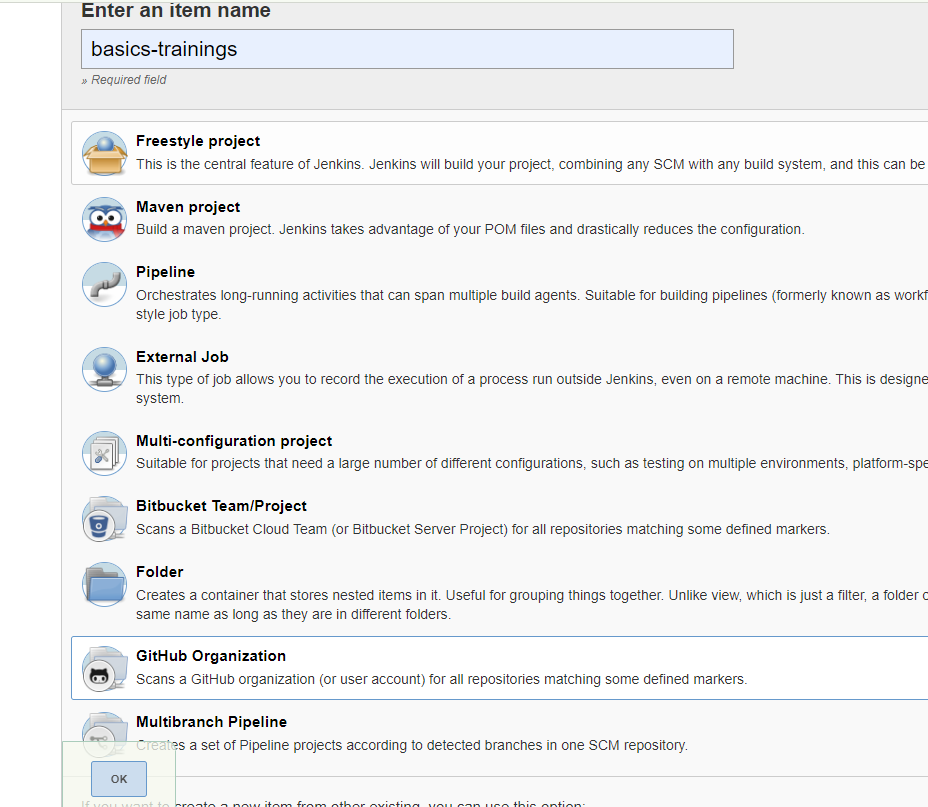
1. Fill the below values and click on OK. Make sure you have username and id as same.



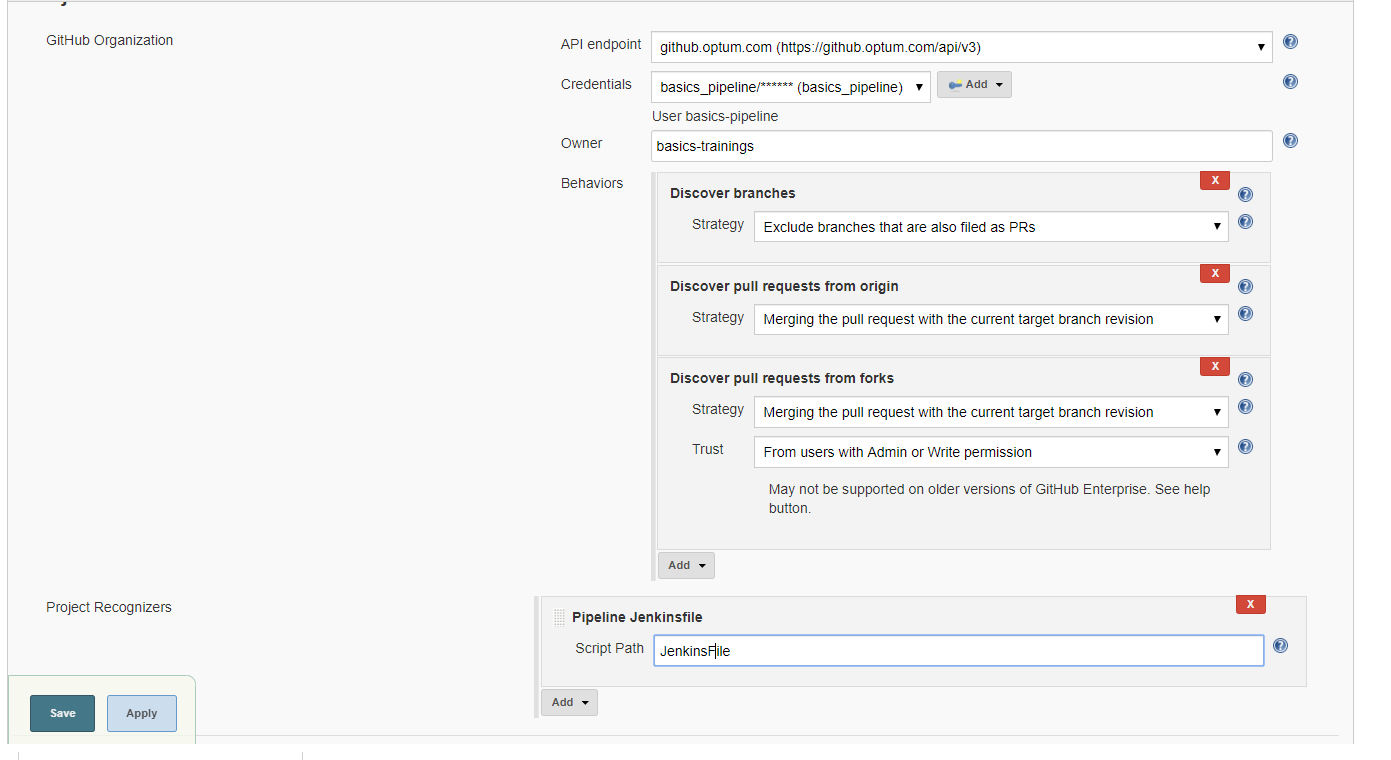
1. Configure github organization in Jenkins
2. Login to Jenkins with your openshift credentials : <https://jenkins-basics.ocp-ctc-core-nonprod.optum.com/>
3. Add Organization Folder –
   1. Click on new Item



* 1. Enter your organization name and select Github Organization and click on Ok

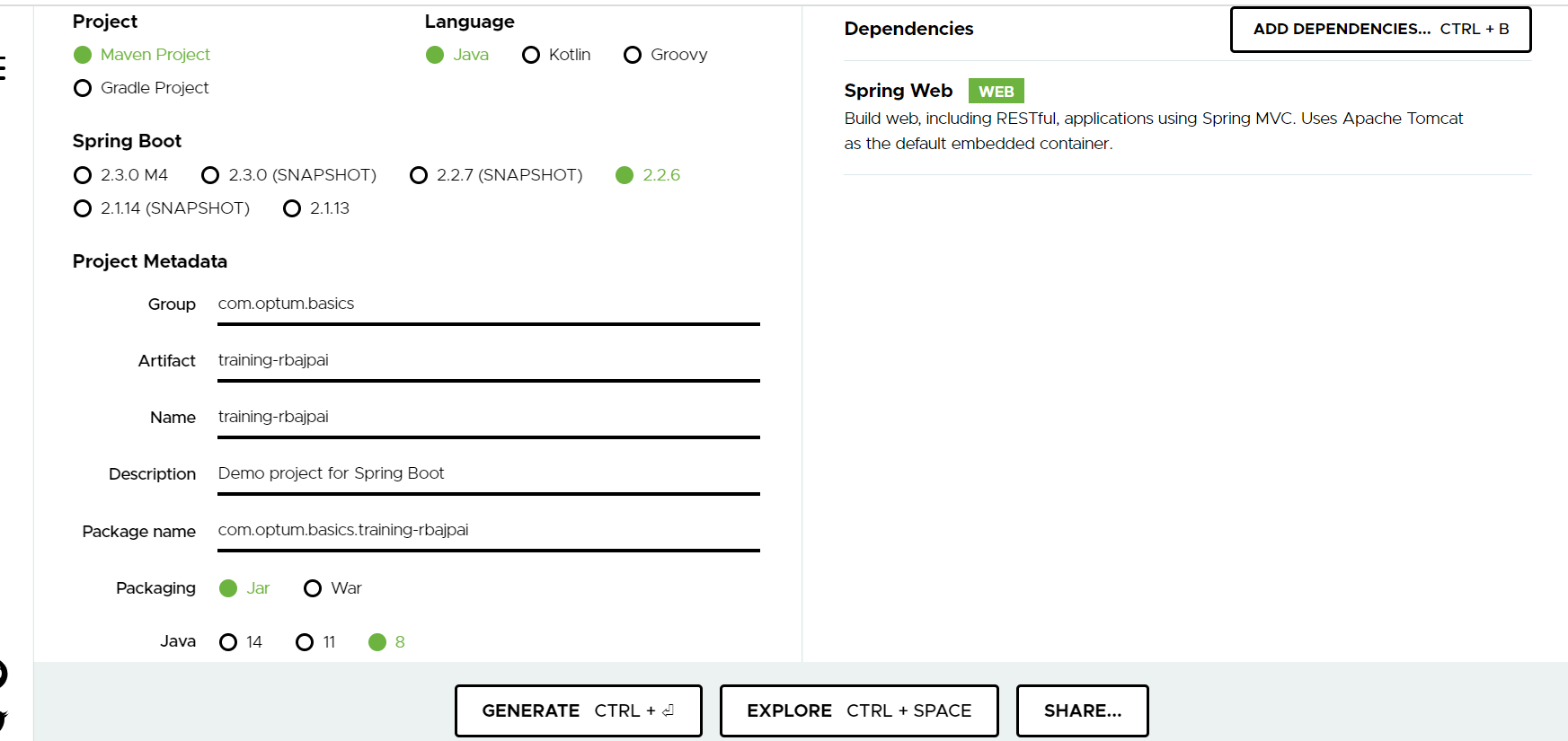


* 1. Select values as per the image below :
     1. Credentials : Select your msid from dropdown
     2. Owner: Organization Name
     3. API Endpoint: github.optum.com



* 1. This will create a folder with organization name and will scan and execute any repo that is having a JenkinsFile. Also it will auto trigger existing or create a new job as soon as a repo with jenkinsfile is added in the organization.

1. Create a Spring Boot project using <https://start.spring.io/>
2. Fill and Select below values and click on Generate button.



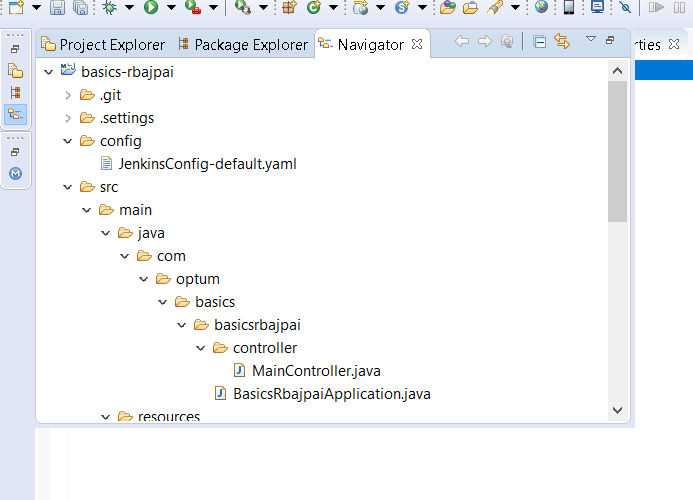
1. A zipped Project will be created and will be downloaded on download folder.
2. Git init
3. Git add .
4. Git commit –m “initial commit”
5. Extract the zipped folder and open as a maven project on your eclipse/STS
6. Adding Context root, Rest API and Actuator.
7. Add following properties in application.properties residing on resource folder.

server.port: 8080

server.servlet.context-path: /yourcontext

spring.application.name: {app-name}

1. Add a controller folder/package and a controller class.



* 1. Make it as a rest controller and a add a basic rest method/api. You can take reference from the attached file.



1. Adding Actuator – Add below dependency on your pom.xml
2. <dependency>
3. <groupId>org.springframework.boot</groupId>
4. <artifactId>spring-boot-starter-actuator</artifactId>
5. </dependency>
6. Run you app locally and see if your application works –

Api URL: <http://localhost:8080/sboot/api/helloworld>

Actuator URL: <http://localhost:8080/sboot/actuator>

1. Adding docker file to root directory of you application.

This will dockerize you application and will create a docker image that will be used in openshift deployment.



* 1. Replace <your\_emailid> with your emailed.
  2. Replace <app-name> with your app name.

1. Adding Jenkins file and optumfile.

This will add Jenkins capability to your application and will do openshift deployment



Replace <app-name> with your app name.



Replace <yourcontext> with your context.



Replace <appName> with your app name.

Replace <OPENSHIFT\_CREDENTIALS\_ID> with your msid.

1. Adding application to git repo
   1. Open GIT CMD.
   2. Change directory to your project root.
   3. Execute following commands
2. git init
3. git add .
4. git commit -m "first commit"
5. git remote add origin [https://github.optum.com/<org>/<app-name>.git](https://github.optum.com/%3corg%3e/%3capp-name%3e.git)

https://github.optum.com/AB/training-boot.git

1. git push -u origin master

This will add your application github.

1. **A job will be automatically created under following location** : [https://jenkins-basics.ocp-ctc-core-nonprod.optum.com/job/<your-organization](https://jenkins-basics.ocp-ctc-core-nonprod.optum.com/job/%3cyour-organization)>
2. **API will be deployed in openshift** - <https://ocp-ctc-core-nonprod.optum.com/console/project/basics/overview>
3. Prometheus Metrics from Spring Boot :
   1. Add Following dependencies in your POM.xml

<!-- Micormeter core dependecy -->

        <dependency>

            <groupId>io.micrometer</groupId>

            <artifactId>micrometer-core</artifactId>

        </dependency>

        <!-- Micrometer Prometheus registry -->

        <dependency>

            <groupId>io.micrometer</groupId>

            <artifactId>micrometer-registry-prometheus</artifactId>

        </dependency>

* 1. Add following properties in application.properties

#============================================================================

#########################ACTUATOR############################################

#============================================================================

management.endpoint.metrics.enabled=true

management.endpoints.web.exposure.include=\*

management.endpoint.prometheus.enabled=true

management.metrics.export.prometheus.enabled=true

* 1. Run Your application locally and see if the following url works : [http://localhost:8080/<context>/actuator/prometheus](http://localhost:8080/%3ccontext%3e/actuator/prometheus)
  2. Commit and push your changes. This will automatically deploy your code.

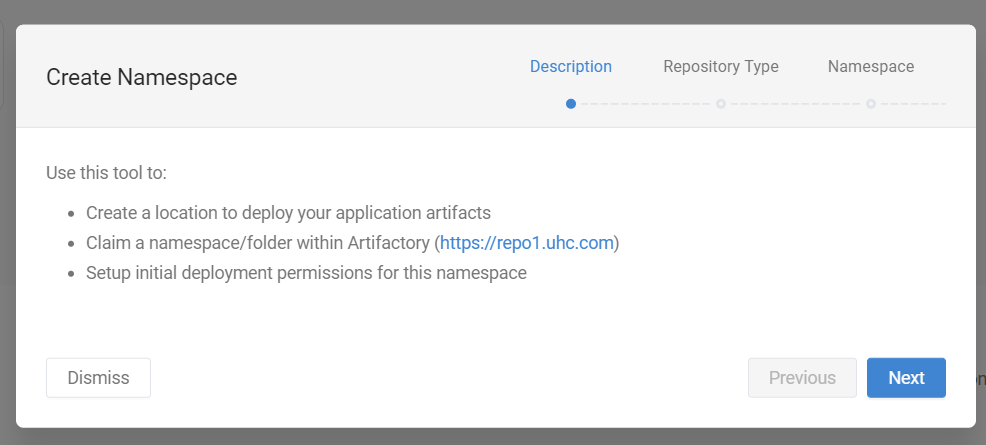
Appendix

1. Stargate Implementation - <https://stargate-docs.optum.com/stargate-self-service-resources/>
   1. <https://github.optum.com/APIGateway/stargate-self-service> fork the stargate repo .
   2. Create namespace

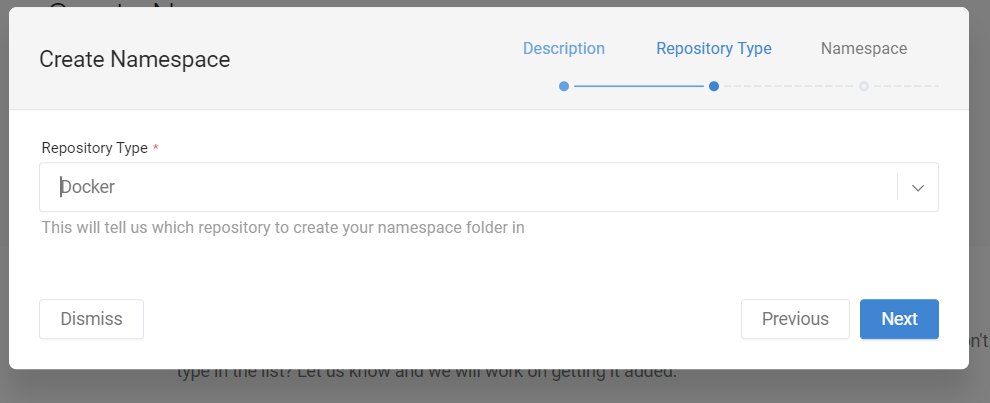
namespace": "basics"

* 1. Create stargate proxy under the requests folder of the above repo.
     1. Checkin the proxy and create the PR.
     2. Its self service and if there are no issues the proxy will be created and email sent to user.

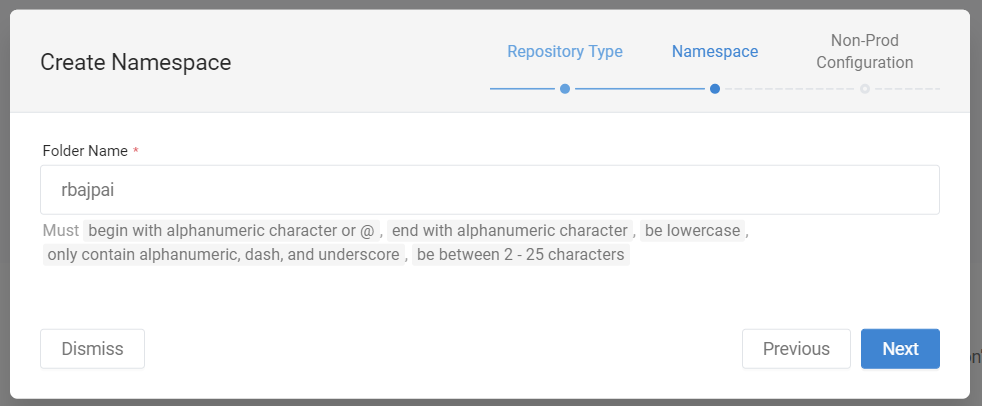
1. Create Docker Repo Namespace for storing docker image
2. Open the link - <https://tech.optum.com/products/artifactory/create-artifactory-namespace> and Login
3. Click on Start Now
4. Click on Next



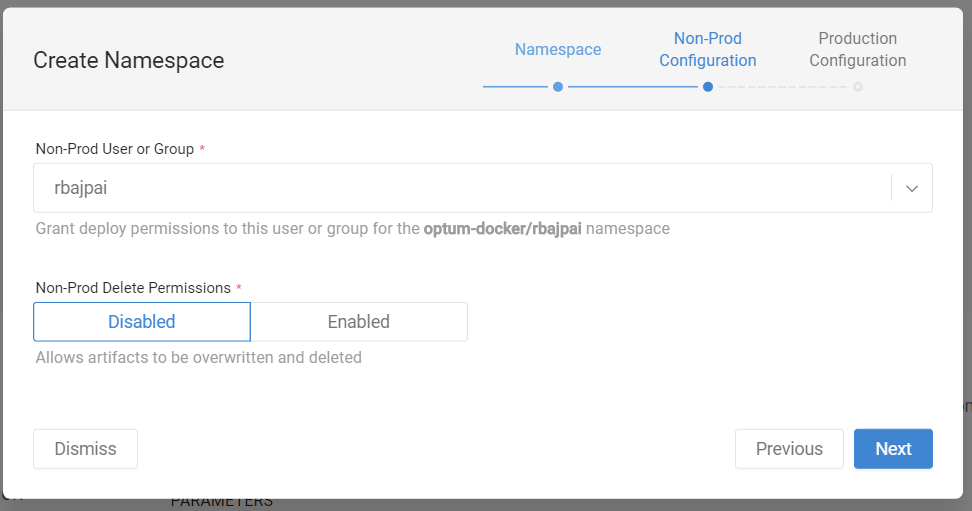
1. Select Docker from the dropdown and click on next



1. Add the namespace you would want to create. (For training please add your MSID) and click on next



1. Add your msid and click next.

7. 

1. Click next again
2. Click on Submit.
3. You can verify by checking your msid on the link - <https://repo1.uhc.com/artifactory/optum-docker/>