**Jenkins Role-Based Access**

As an example in your organization want to create new roles with a set of a specific person to access Jenkins and assigning roles to individual users. The [Role-based Authorization Strategy plugin](https://plugins.jenkins.io/role-strategy) is designed to provide this kind of functionality. This plugin allows three type of functionality

*Global roles —*across a project with permission job, run and Source Control Management (SCM)

*Project role — c*an access particular project job or run category

*Slave role —*just view the project with permission to admin nodes

Go to Manage Jenkins>> Manage Plugin, Available tab and search bar type “role”

If you have done this installation go, Manage Jenkins,>> Configure Global Security if security is enabled in Jenkins and you can see a new option named Role-Based Strategy under Authorization in the Access Control section

Graphical user interface

Description automatically generated

If this option is selected and saved, there will new option available in Manage Jenkins page named “Manage and Assign Roles”

A screenshot of a computer

Description automatically generated with low confidence

Manage and Assign Roles screen are three selections for the main functions: “Manage Roles”, “Assign Roles”, and “Role Strategy Macros” . This blog i will cover “Manage Roles”, “Assign Roles”.

Graphical user interface, text, application, email

Description automatically generated

Before you assign roles you have to create users Manage jenkins>> Manage Users

A screenshot of a computer

Description automatically generated with medium confidence

I create some users call dev,qa, and intern. Dev can access all project and qa can access deployment project, an intern can view the project only basic project can access.

Graphical user interface, application, Teams

Description automatically generated

**Manage Roles**

As the name suggests, this screen allows you to create or delete roles and assign them permissions. There are three sections here for each of the three roles listed above: Universal, Project, and Slave. Go to “Manage Role “ and I create two “Global roles”

Table

Description automatically generated

Here you can simply type your role name in the “Role to add” hit the add button and it will add on global roles. After adding the role, you can check the appropriate boxes to give the role the desired permissions. If you need to create new user overall read permissions just do the same way and check the “Read” box.

“Project roles” we can give the project base authentication(Jenkins job base ), we have ASP.net Core environment so in my case i add some project role base on the project pattern. we can select permissions for the role just as we did for the global.

Graphical user interface, application, Word

Description automatically generated

This case the only tricky field is the Pattern one. For example, if you set the field to “dotnet.\*” then the role will match all jobs which name starts with “dotnet” and that is case-sensitive. perform a case-insensitive match, use (?i) , upper case “Dotnet.\*” lower case “dotnet” jobs can user this pattern “(?i)dotnet.\*” . I create three roles and three type jobs

Table

Description automatically generated

save all configuration and verify configure just going again “Manage Roles” .

**Assign Roles**

Graphical user interface, text, application

Description automatically generated

Once we have our desired roles set up, we can assign users or groups to particular roles

Graphical user interface, text, application

Description automatically generated

type user firstly we created and hit the “Add” button . You can check the appropriate boxes to give the role the desired permissions. “Item roles” and “Node roles.” To be clear, “Item roles” here corresponds to “Project roles” and “Node roles” corresponds to “Slave roles.”

Graphical user interface, text, application

Description automatically generated

save all configuration and verify configure just going again “Assign Roles” . All are done now and you can see my project . Here i create Free style jobs case-sensitive and case-insensitive way “Dotnetproj”, “dotnetproj-emp” and “dotnetproj-deploy” these jobs can see “dev” users . “dotnetproj-deploy” can see “qa” users and “sample-dotnet-emp” can see “intern” users

Admin Login

Table

Description automatically generated with medium confidence

Dev Login

A picture containing graphical user interface

Description automatically generated

case-sensitive and case-insensitive way “(?i)dotnet.\*”

QA Login

A picture containing graphical user interface

Description automatically generated

Intern Login

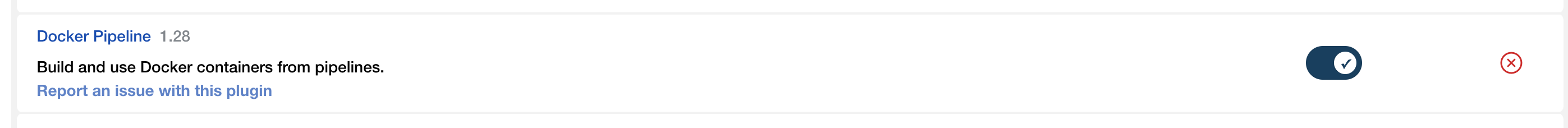
Graphical user interface

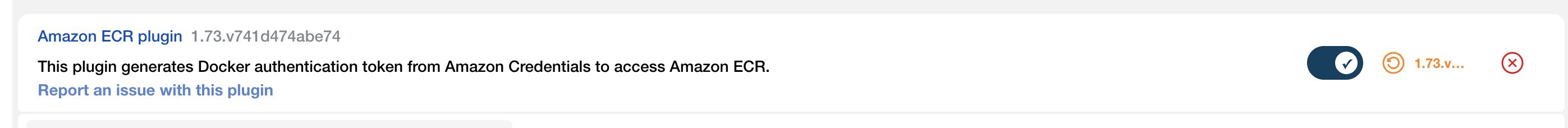
Description automatically generated with low confidence

Docker build

You can store your Amazon AWS credentials in Jenkins using the Amazon ECR [plugin](https://plugins.jenkins.io/amazon-ecr/), and you can use them by using the [Docker Pipeline](https://plugins.jenkins.io/docker-workflow) plugin

Install them





Aws ECR plugin offers integration with [Amazon Container Registry (ECR)](https://aws.amazon.com/ecr/) as a [DockerRegistryToken](https://github.com/jenkinsci/docker-commons-plugin/blob/master/src/main/java/org/jenkinsci/plugins/docker/commons/credentials/DockerRegistryToken.java) source to convert Amazon Credentials into a Docker CLI Authentication Token.

Docker pipeline Jenkins plugin which allows building, testing, and using Docker images from Jenkins Pipeline projects.

Once these installed, In your dashboard, add the AWS credentials.

Manage Jenkins -> Manage credentials -> AWS credentials [kind]

Graphical user interface, text, application, email

Description automatically generated

**Credentials**

Set below for further

Table

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application, email

Description automatically generated