Documentation on BitBucket Pipelines

Components:

`test`, `staging`, or `production` Its, stating of the pipelining define the build process of the repo which we are want to create a pipeline.

```
- step:
   A group of execution can be consider as a step. We can define no.of steps based on the creation on pipeline.
Default:
        Common tag which execute are every push on to the repo. If we want to stop triggering we can specify the [skip ci ] in the commit tag.
       Branches: which applies to all branch level creation.

Pull-requests:
        triggers on applying pull on the repo branch. If we ant to make trigger to    all branches on pull request we use in place of the branches.
Tag:
       tag specific build pipelines in git hub
       '*-linux':
            - step:
                name: Build for *-linux tags
                script:
                  - echo "Hello, Linux!"
       The tag should match with the tag in git repo.
Custom:
      this property are getting to triggered on manual.

Parallel:
      It’s a property which can run set of steps under it together.

       Pipelines:
          Default:
             Parallel:
                Steps:
                 -step:
                    Name: parallel step
                    Script:
                        Echo “I am parallel”
                  -step:
                     Name: parallel
                     Script:
                        Echo “I am in parallel”
```

```
    Branches:
         Master:
             -step:
                Name: non parallel
                Script:
                    -  Echo ”non-pararlle”
 Staging:
        It is used to group few steps inside the staging for execution.
        which beneficial to make clear execution.

        pipelining:
            defaults:
            staging:
                name: build and deploy
                steps:
                            -  Step:
                                  Name: building
                                  Script:
                                        -./build-app.sh
                            - step:
                                  Name: deploy
                                  Script:
                                        -./deply-app.sh
            branches:
            pull-requests:
            tag:
            custom:

deployment:
        Make the environment for the deployment set as test , staging and
       production
       pipelining:
          default:
             -stag:
                 name: staging
                 steps:
                    -step:
                       Name: build
                       Script:
                           Echo “building step”
                    -step:
                        Name: testing
                        Script:
                            Echo: “testing step”

          branches:
          pull-requests:
          tag:
```

name:
used in step, stage and pipe

triggers:
automatic and manual and it are on parent of stage, step

step:
This property is used to define the execution unit. They executed in the ordered way.
Each step in yam file are executed in a separate container with followed script.
They have the properties we can use with it like:
Image:
to run a different image.

Services: specific defined service

Caches: specific defined cache

Artifacts: to retain the artifact for following steps to use

Trigger: make it manual/automatic.

Clone: manual clone to the repo.

Parent properties are – (default, parallel, custom, branches, tags, steps)
Child properties: (name,script,after-script,image,services,deployment,trigger,caches,clone,artifacts,conditions)

Script:
we can make set of cmds execution in steps.
Parent: (step)
Child: (pipe)

Size:
which place to increase the size of memory for services and script. 1x default and 2x double the previous.

After-script:
which is to execute after the script property is completed. Note : if any of the property in the after-script is failed then there wont affect to the final out come.

Parent- (step)
Child- (pipe)

Name:

this property is displayed in the pipeline log. It for stage,step,pipe,

Fail-fast:

this property is used to set up in steps execution. If , its is : false then it wont effect to the parallel group step if it set :true then whole steps are halted.

Parents: (step, parallel)

Artifacts:

which contain the build artifacts for project execution.
Downloads: true/false
Paths:

Pipes:

they make the complex properties more easier by passing the requires variables if necessary.
We can create a custom pipes and use them .

Runner:

Condition:

Which is to set and only on meets the condition. It will execute the step.
Child(changesets, includepaths)

Service:

This property is used then bit will create a separate container execution for the service which make the execution fast.

Deployment:

Place the environment for stage or step for execution. defaults environments are staging, test, production for these properties.