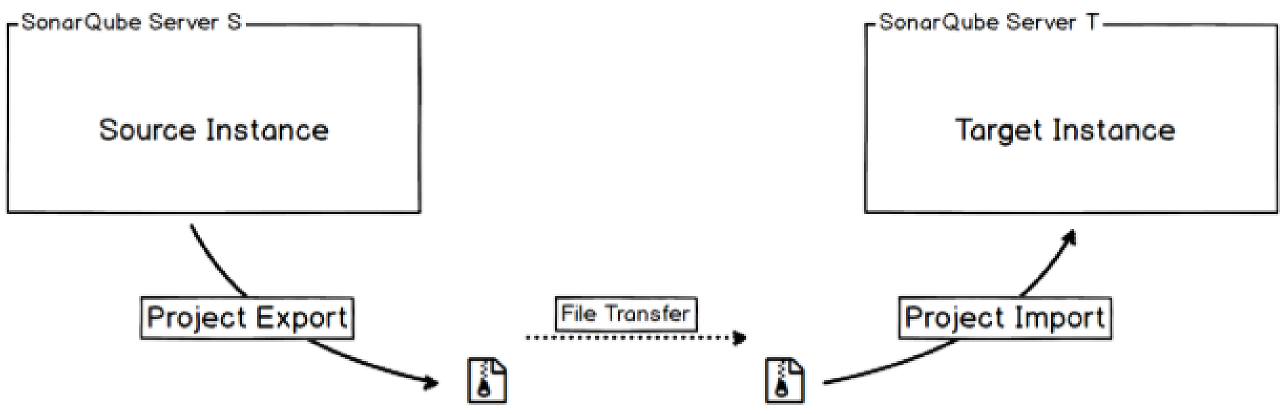
Import & Export configuration for sonar



1.Using project move

Project Move is available as part of Enterprise Edition and above.

Project Move allows you to export a project from one SonarQube instance and import it into another, identically configured SonarQube instance. To use Project Move, you must have the Administer permission on the project in the source instance, and access to the file systems of both instances.

When to Use ProjectMove

- you want to create a central SonarQube instance at enterprise level and you want to keep the history created on N instances used previously at the team level.
- One company is acquiring another company that already has a central SonarQube instance.

Prerequisites

- the exact same version
- the same plugins with the same versions
- the same custom metrics
- the same custom rules

How To Export

- Navigate to the project and at the project level, choose **Administration > Import / Export**
- Click on Export button to generate zip file containing setting and history of project.
- A zip file containing all project data ex is generated in *$SONAR*SOURCE*HOME/data/governance/project*dumps/export/_ named *<project* key>.zip_

How To Import

- With a user having the "Administer System" and "Create Projects" permissions, go to **Administration > Projects > Management** and **Provision the project** using same project key
- Configure the Project's permissions, and the Quality Profiles and Quality Gate associated to the Project
- Put the generated zip file into the directory *$SONAR_TARGET_HOME/data/governance/project_dumps/import*
- Go to the Project's Home Page and choose **Administration > Import / Export**
- Click on the Import button to start importing your data
- Once the import is finished, trigger an analysis to import source files into the new instance.

2.Using Rest API

To create quality gate using API

```
curl -u admin:admin -X POST
'http://novartis.devops.altimetrik.io:9000/api/qualitygates/create?name=
NOV'
```

And create condition

```
curl -u admin:admin -X POST
'http://novartis.devops.altimetrik.io:9000/api/qualitygates/create_condi
tion?gateId=1&metric=blocker_violations&op=LT&warning=5' --create
quality gate
```

Automation script

Python script to import configuration from one server to another server using REST API

```
import os
import subprocess,json,requests
from ast import literal_eval
import json
import pdb
#pdb.set_trace()
data =
requests.get('http://novartis.devops.altimetrik.io:9000/api/qualitygates
/list', auth=('admin','admin' )).json()
print (data)

#for name in data.values():
 #   print (name)

b=data.get ('qualitygates')
for a in b:
    c=a.get('name')
    show =
requests.get('http://novartis.devops.altimetrik.io:9000/api/qualitygates
/show?name='+c, auth=('admin','admin' )).json()

print("*****************************************************************
**")
    print (show)

print("*****************************************************************
**")
    id = show.get('id'))
    params = (
        ('id', '7'),
    ('name', 'tes'),
    )
    response =
requests.post('http://novartis.devops.altimetrik.io:9000/api/qualitygate
s/copy', params=params, auth=('admin', 'admin'))
```

Updated Python script using REST API

GIT REPO: code is available http://novartis.devops.altimetrik.io:7080/root/sonar-config-migrator

Branch:feature_sonar

```
import os
import subprocess,json,requests
from ast import literal_eval
import json
import pdb
from common import config


#print(config.credentials,config.source_url)


#print('source',config.source_url)
data = requests.get( config.source_url1, auth=config.credentials
).json()
#print (data.text)

#for name in data.values():
#   print (name)

get_qualitygate=data.get ('qualitygates')
for name in get_qualitygate:
    qualitygate_name=name.get('name')
    show = requests.get(config.source_url2+qualitygate_name,
auth=config.credentials ).json()
    params = (
    ('name', config.name),
    )

    response_gate = requests.post( config.remote_url1 , params=params,
auth=config.credentials ).json()
    print(response_gate)
    params = (
    ('id', 'response_gate'),
    )
    response = requests.post( config.remote_url2 , params=params,
auth=config.credentials )
    params = (
    ('gateId', response_gate['id']),
    ('metric', 'blocker_violations'),
    ('op', 'LT'),
    ('warning', '5'),
    )

    response = requests.post( config.remote_url2 , params=params,
auth=config.credentials )
    params = (
    ('gateId', response_gate['id']),
    ('metric', 'critical_violations'),
    ('op', 'GT'),
    ('warning', '5'),
    ('error', '10'),
    )

    response = requests.post(config.remote_url2 , params=params,
auth=config.credentials )
    params = (
    ('gateId', response_gate['id']),
    ('metric', 'test_errors'),
    ('op', 'GT'),
    ('warning', ''),
    ('error', '20'),
    )

    response = requests.post(config.remote_url2 , params=params,
auth=config.credentials )
    params = (
    ('gateId', response_gate['id']),
    ('metric', 'test_failures'),
    ('op', 'GT'),
    ('warning', ''),
    ('error', '20'),
    )

 response = requests.post(config.remote_url2 , params=params,
auth=config.credentials )
 params = (
 ('id', response_gate['id']),
 )

    response = requests.post(config.remote_url3, params=params,
auth=config.credentials)
    import logging
    logging.basicConfig(filename='quality.log',level=logging.DEBUG)
   # logging.info('response code ='+str(response.status_code),'response
body = '+ str(response.json()))
```

```
    #print('response code ='+str(response.status_code),'response body = 
'+ str(response.json()))
```

Configuration python file

```
credentials = ('admin','admin')
source_url= 'http://novartis.devops.altimetrik.io:9000/'
dest_url= 'http://3.92.127.4:9000/'
endpoint1 = 'api/qualitygates/list'
endpoint2 = 'api/qualitygates/show?name='
endpoint3 = 'api/qualitygates/create'
endpoint4 = 'api/qualitygates/create_condition'
endpoint5 = 'api/qualitygates/set_as_default'
source_url1 = source_url + endpoint1
source_url2 = source_url + endpoint2
remote_url1 = dest_url + endpoint3
remote_url2 = dest_url + endpoint4
remote_url3 = dest_url + endpoint5
name = 'NOV'
```