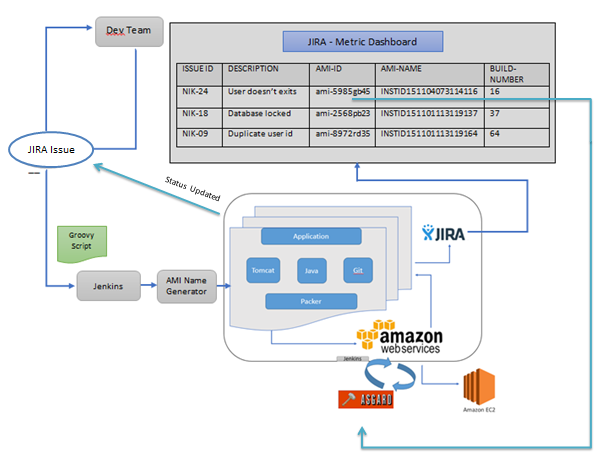
**Nike POC**

Pre requested Software’s:

1. JIRA 7.X
2. AWS account
3. Github account
4. Jenkins 1.6X
5. Packer 0.8X
6. Asgard 1.5.X
7. Webserver

**Project flow**



**Steps:**

1. Create a custom workflow in JIRA.
2. Create a custom task type and map custom workflow with that issue type.
3. Create an issue in JIRA.
4. For every issue under our issue type will come under our custom workflow.
5. Dev team will replicate the issue, and start fixing the bug.
6. Dev team commits their code in Github with Issue id and Commit Message.
7. Commits will be replicated in corresponding issue in Jira if you installed Git plugin in Jira.
8. Trigger Jenkins job from JIRA issue page.
9. Through Groovy script we will get the issue id in Jenkins.
10. Here we have created 4 Jenkins job for this scenario
11. 1st job – Will get the Issue ID. & it will generate Unique AMI Name and push into the 2nd job
12. 2nd job will initiate the packer to generate instance, and it gets the latest code from Github and start building the application, once build is success it will deploy the application in server, then packer will build AMI using the EC2 instance.
13. 3rd job will analyze the 2nd job log and get the AMI-ID generated by packer and stored in the property file.
14. 4th job will update all the build information, AMI-ID, AMI Name and push back into the JIRA through Rest Api.
15. Created a customized dashboard using Gadgets to show all the necessary fields.
16. Download and Deploy Asgard war in server directory, and run the server with JAVA\_OPTS.

Or

Follow the below server configuration steps mentioned in the below link.

<https://github.com/Netflix/asgard/wiki/Tomcat-configuration>

1. Create an application in the Asgard, and configure AMI, Security group, ASG.
2. Once it create successful. It will start the instance in AWS.
3. Run the deployed application from the New Instance.

**Sample rest service to update the values in JIRA**

**Application/Json:** Use JIRA rest update service with below Json as input.

{

"update": {

"FieldName": [

{

"set": Value

}

]

}

}

**Packer:**

Packer tool which help to build our application code with base image and generate AMI or Virtual box or Vagrant image.

**Sample Packer Json file:**

{

"variables": {

"aws\_access\_key": "",

"aws\_secret\_key": ""

},

"builders": [{

"type": "amazon-ebs",

"access\_key": "{{user `aws\_access\_key`}}",

"secret\_key": "{{user `aws\_secret\_key`}}",

"region": "us-east-1",

"source\_ami": "ami-de0d9eb7",

"instance\_type": "t1.micro",

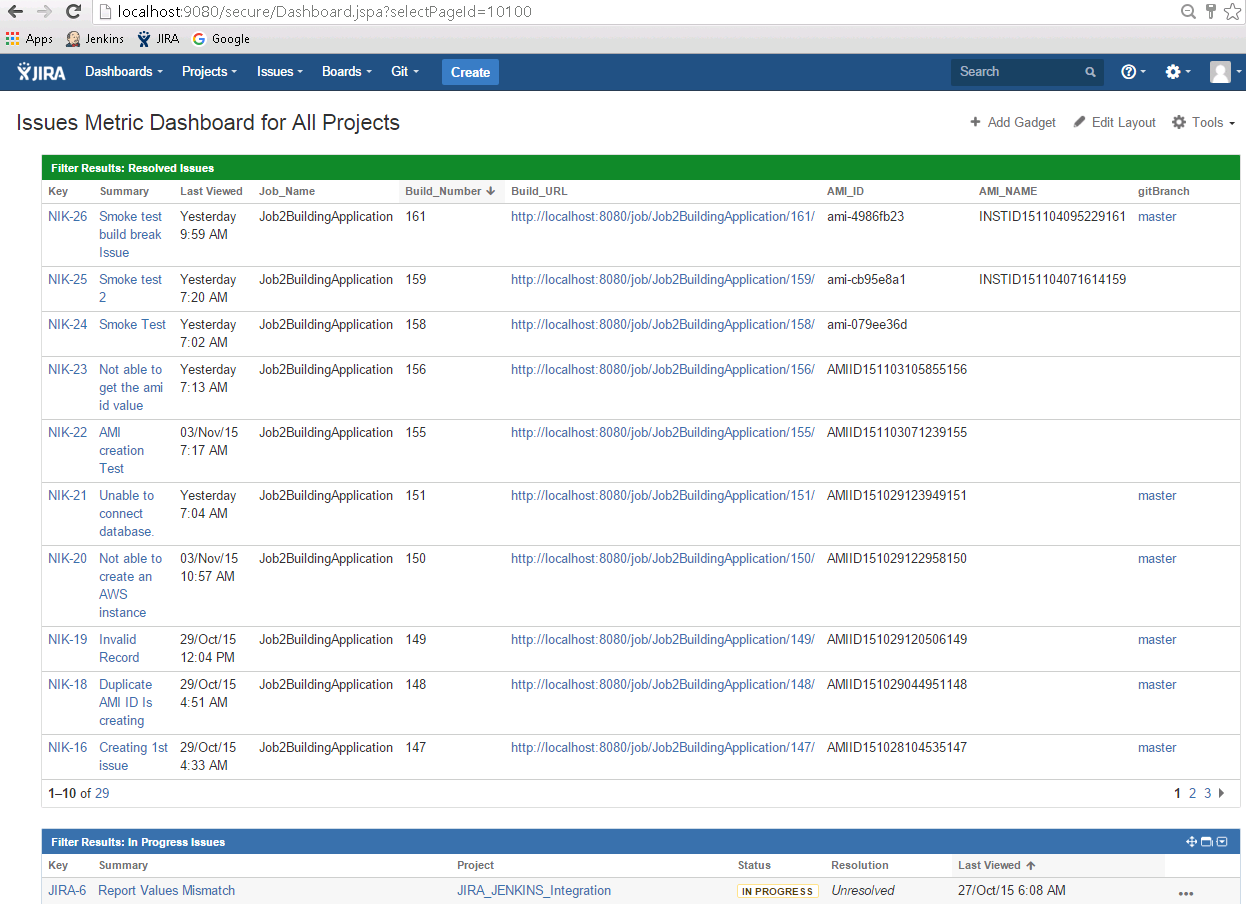
"ssh\_username": "ubuntu",

"ami\_name": "packer-example {{timestamp}}"

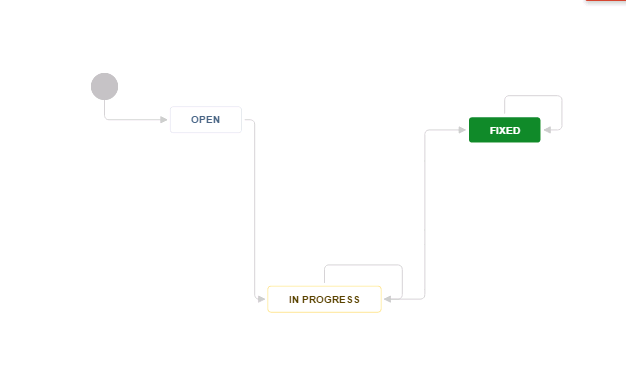
}]

}

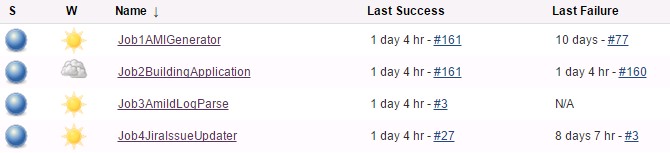
**Screenshot: Customized Dashboard**



**Custom workflow:**



**Jenkins Job:**



**Sample Groovy script to pass the issue id to Jira:**

Issue issue = issue;

def id=issue.getKey();

def url = "http://10.0.0.75:8080/job/Job1AMIGenerator/buildWithParameters";

println "C:/Program\_Files/cURL/bin/curl.exe -X POST $url?JIRA\_KEY=$id".execute().text;

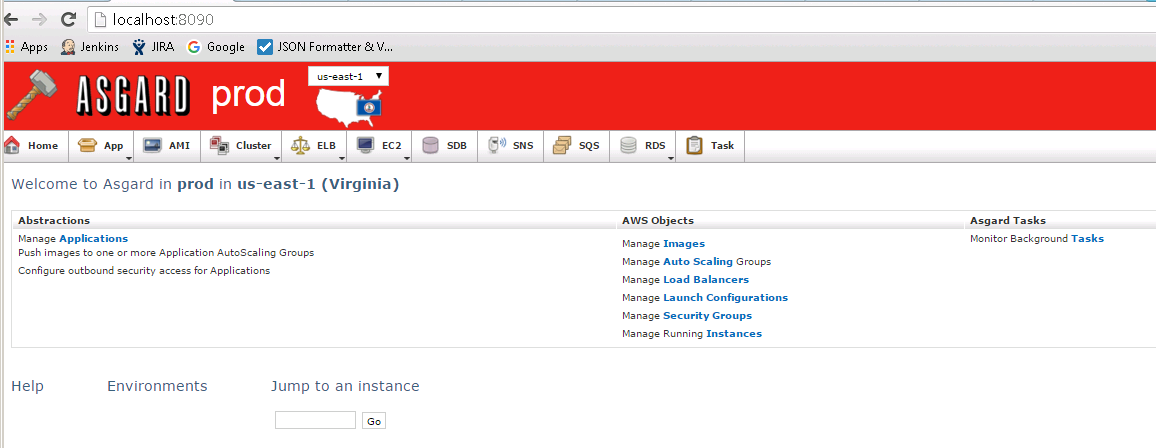
**Exporting Issue information from JIRA as Rest service.**

**Sample Rest service:**

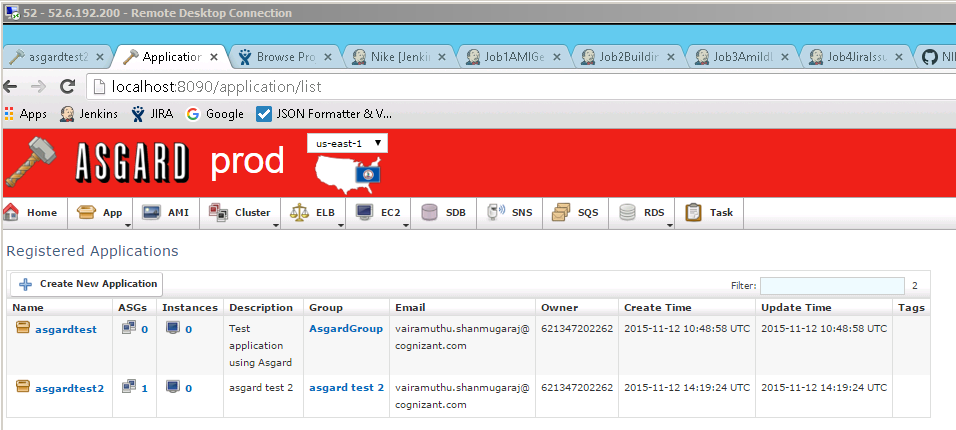
http://localhost:9080/rest/api/2/search?jql=issuetype = Task AND AMI\_NAME ~ INSTID151104095229161

**Asgard:**

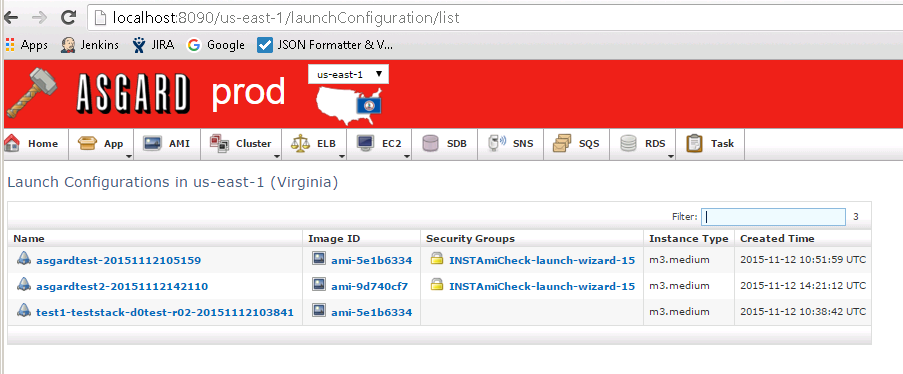
Asgard application Home page:



**Application List:**



**Launch Configuration list:**



**Instance running in EC2 from Asgard:**

