RAJESH PULABALA

1. Application (a simple ruby app that responds with "Hello World!" to anyone who visits it) is reasonably resilient and a single node failure does not affect end users. You can either use the source code provided in this repo or create your own application.

I took one Java sample program “Hello world”. That is

class Sample{

public static void main(String args[]){

System.out.println("Hello World");

}

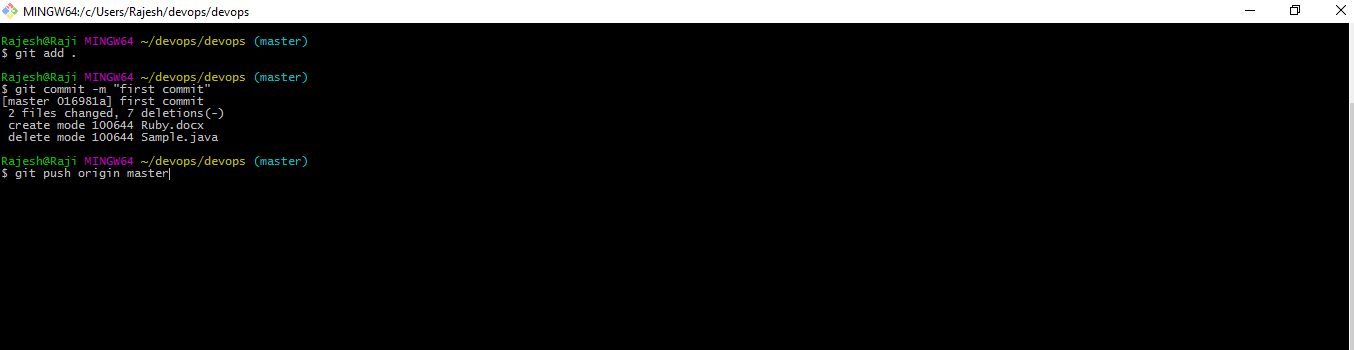
}

For example if this is the code developed by the Developer. And push into git repository by using git commands.

git add <file name>

git commit –m “give commit message”

git push origin master.



Once a developer push the code then I will trigger a build by using web hooks or else we can use git plugin in Jenkins that is git is integrated with Jenkins. So automatically our source code from the repository will come into Jenkins works place. In that build process we are going to be invoke build.xml file which is use to make that source code into jar/war depends on requirement.

Build.xml

<project name="first project" default = "run">

<target name = "compile">

<echo> compling the code </echo>

<mkdir dir ="build/classes"/>

<javac srcdir="source" destdir="build/classes"/>

</target>

<target name = "package">

<echo> packaging the code </echo>

<mkdir dir ="build/jar"/>

<jar basedir ="build/classes" jarfile="build/jar/Helloworld.jar">

<manifest>

<attribute name="Main-class" value=" Helloworld "/>

</manifest>

</jar>

</target>

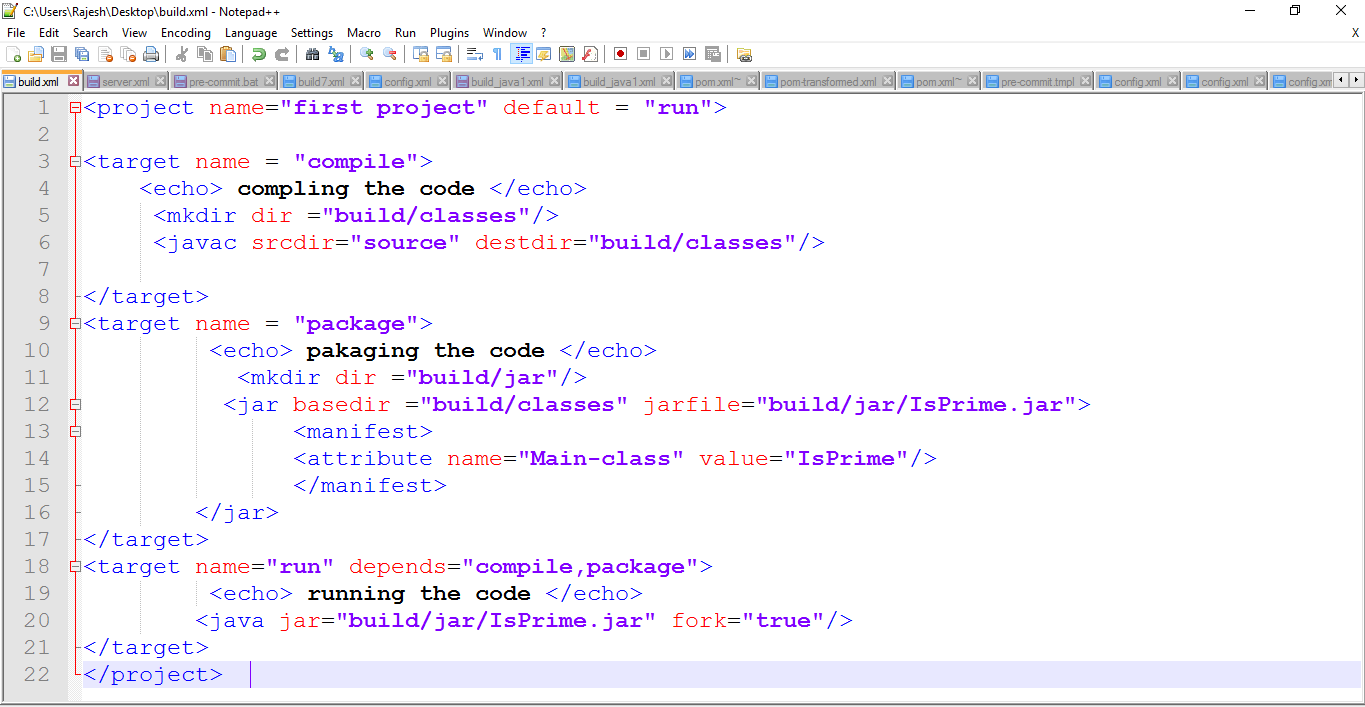
<target name="run" depends="compile, package">

<echo> running the code </echo>

<java jar="build/jar/IsPrime.jar" fork="true"/>

</target>

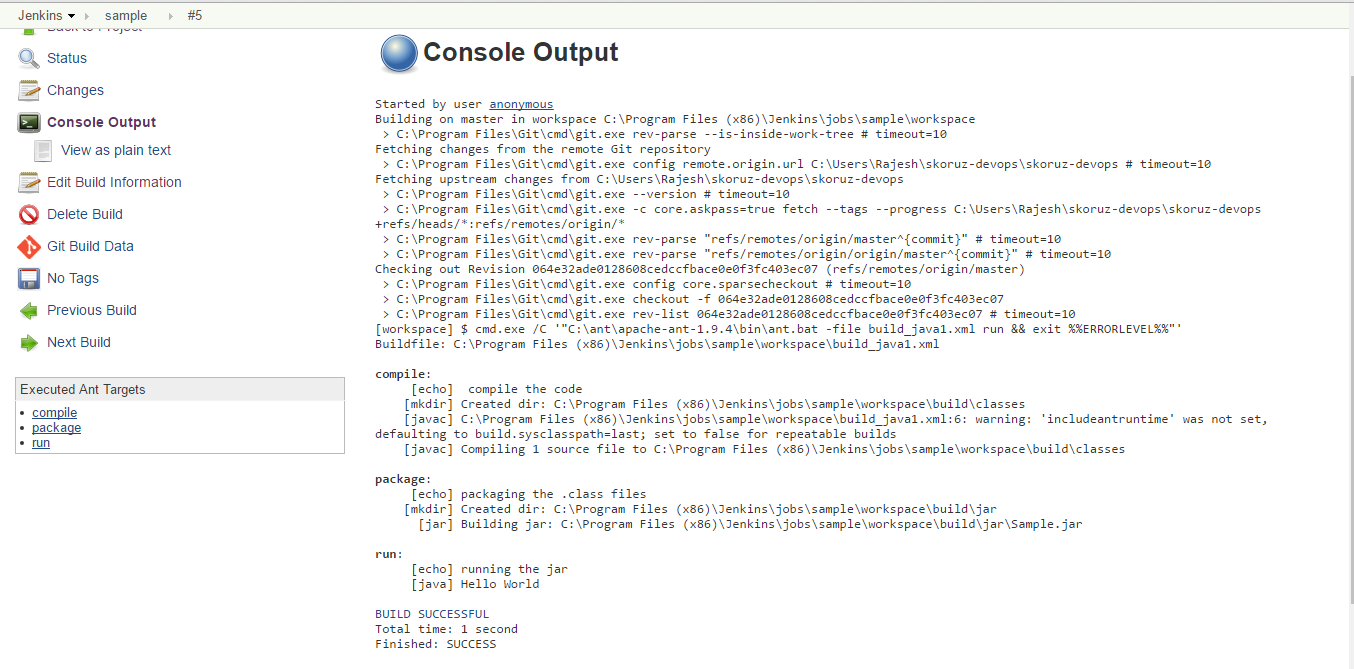
</project>



sample code I am going to do compile and package and run by using the build tool ANT (or) we can use another build tools like MAVEN/Gradle and using CI tool like Jenkins I can make this as jar/war. And then going to deploy it on Application server like Tomcat server.

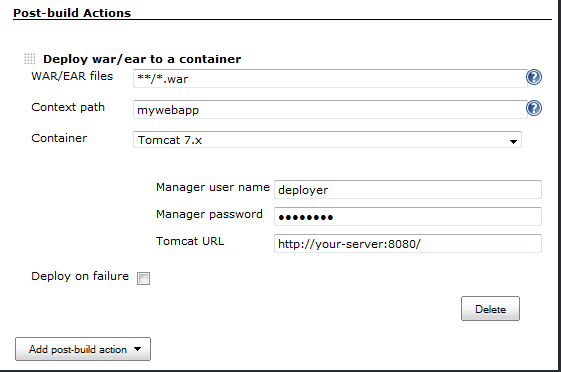
Jenkins:-

Now I will create a job inside Jenkins once I configure everything inside that job now I will invoke build.xml in invoke ant step.then I will run that build.i got console output like this.



So now I got war file if you clearly observe from the output. So now I can able to deploy that war file

In Tomcat server.



For example if we install tomcat server in amazon ec2 instances. Then we can deploy by using post build actions.

Once if we deploy our application into the tomcat server then we can able to run the application in localhost.

Application can be scaled, preferably automatically, to handle increased loads and as well as Infrastructure and required services provisioning as well as application deployment is automated and can be triggered with a click of a button or a command in a terminal.

So now I am going to setup for adding chef to our auto scaling instances. Chef is a configuration management tool and we will build out an environment that utilizes chef server, s3, and auto scaling to deploy our instances.

One of the core component of the auto scaling is being able to deploy out a new AMI. It should be able to handle any increase in demand. This is called Elasticity.

NOTE: - If we are using auto scaling property. if instance goes down or unhealthy then new instance will be fired up. Tell me how that new instance can be act same like the old instance if newly created instance will be created in another available zone.

(OR)

EXAMPLE : - if we use auto scale the instance min = 4 and, max = 10. Assume that 10 instances are running an application on tomcat server. so if demand increase more than 10 then will able to provision on-demand instances. Tell me how this 11 instance can be able to act as tomcat server and deploy our application and how our end customers can able to access the data from 11 instance?

This is done by using configuration management tools like chef/ansible/puppet.

For example if we use chef.

1. Initially I will create IAM role because when the new AMI’s starts up it communicates with amazon s3 and down loads latest version of it’s pem key and authentication key in order to new instances to be bootstrap as well as the role that the instance should be assumed. If our role is application server then it will download application server nothing but tomcat .
2. if we want deploy our application into that instance then we will write a chef cookbooks and assign a role to thet instance.
3. So I will create a new role for newly created instance can able to communicate with amazon S3.
4. And I will give full permissions on S3 bucket.
5. Create AMI and add this role to that machine image.
6. So now I can able to bootstrap my new instance because my pem key and authentication can be stored in S3.
7. Once bootstrap was done then I will write a role that each newly created role can be act as webserver/application server. Then it will assign that role to newly created instance.
8. For example if I want to act my instance as a web server then I will write cookbook for download and run web server and upload it to chef server whenever instance created.

In order to run the command line interphase we need to install python pip.

1. Install python pip
2. Pip install awscli

Now we can able to run command line so we will configure aws

1. Copy our validation.pem and client.pem into S3 bucket. Because we assign a role we can able to access S3 bucket.
2. Now we can able to bootstrap the new instance with our chef server.

Now we can create an image from this instance and able to provision Ec2 instance’s in any available zone.

Now we have AMI created.

Create launch configuration.and assign auto scale groups

And we can able to use Elastic load balancer and here we can use cloud watch metrics for the host level so by using this we can define our cpu utilization and swap memory etc.

Conclusion: - finally by using configuration management tool chef I can able to assign a role to that newly created instance. And by using auto scaling, elasticity and scalability we can able to meet the increase in demand by provisioning the on-demand instance in aws.