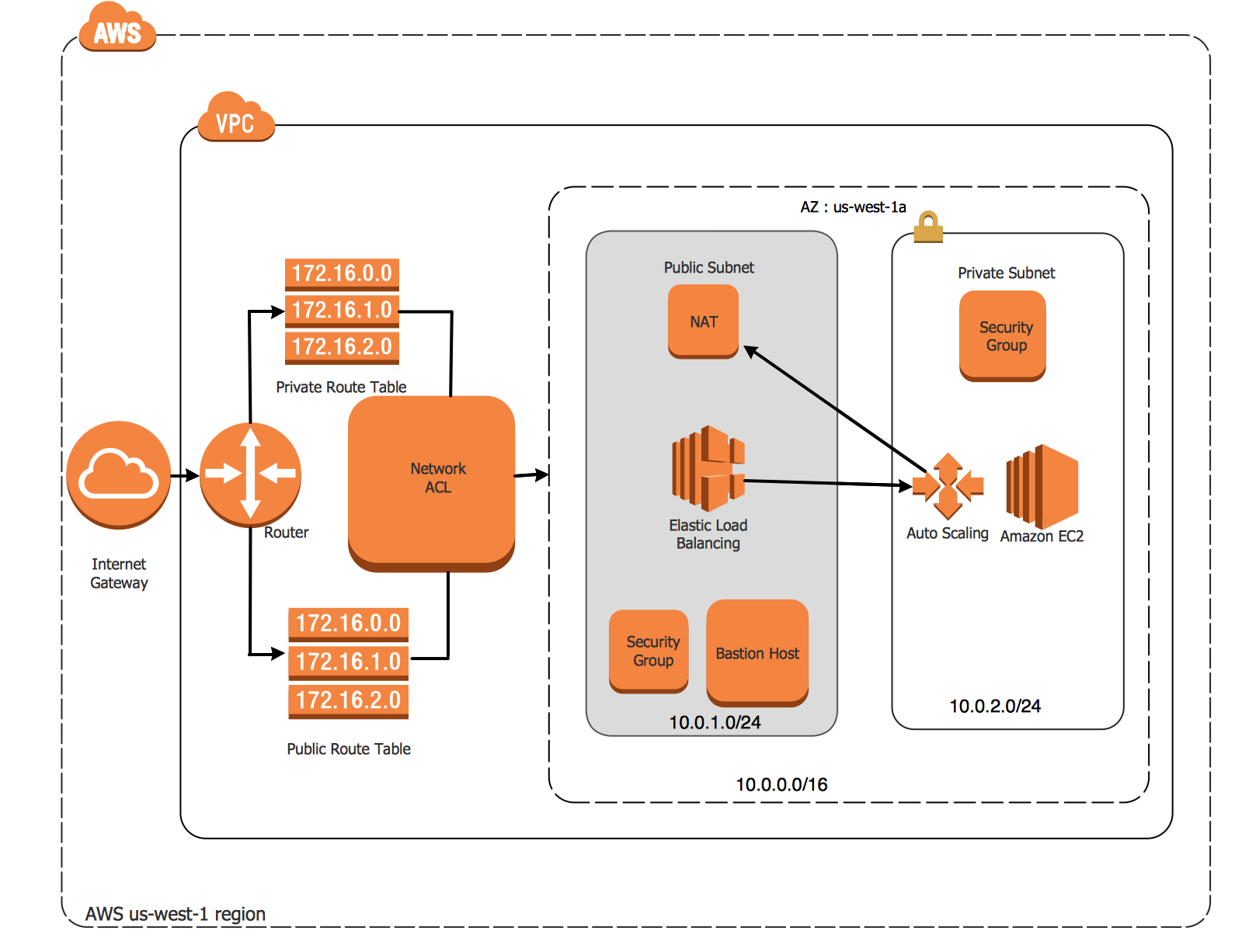
**Want to setup Drools Kie-Workbench and Server in AWS?**

Follow below simple steps to setup Kie workbench and Kie server in AWS.

**Architecture Diagram:**

****

**Important notes to be pointed:**

* You need to attach only public subnets to your ELB, making sure that the availability zones those subnets are aligned with the availability zones of the private subnets that your instances are running.(i.e private and public subnets are in same availability zone like example us-west-1a)
* Make sure that the security group of your instances allows access from the security group of your load balancer
* Make sure that your health check is working locally on the instance. For example, if your health check in the ELB is HTTP:8080/health\_check, on the instance you can curl x.x.x.x:8080/health\_check (where x.x.x.x is the private IP of the instance) and get a 200 response code.
* The public subnet routing table should route 0.0.0.0/0 to the Internet gateway attached to your VPC.
* The private subnet routing table should route 0.0.0.0/0 to a NAT gateway in a public subnet

Nice blog about Bastion Host, NAT gateway: <http://cloudacademy.com/blog/aws-bastion-host-nat-instances-vpc-peering-security/>

**Tools and Technology used:**

Chef-Solo, AWS Cloud Formation templates, AWS management console, Java 7, Tomcat 8, Maven 3+, Drools 6.4.0.Final

**Pre-requisite**

Have Basic knowledge of AWS commands and AWS Management Console.

You should have an AWS account, OR you can create Free Tier AWS account using <https://aws.amazon.com/s/dm/optimization/server-side-test/free-tier/free_np/>

**Step: 1**

*Get Access Key and Secrets key*

Signup for an AWS account (if you do not have one) and get the access key ID and secret access key for your user.

AWS documentation <http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-set-up.html#d0e2883>

**Step: 2**

*Configure AWS CLI*

Install AWS Command line interface (CLI).

AWS documentation <http://docs.aws.amazon.com/cli/latest/userguide/tutorial-ec2-ubuntu.html#install-cli>

**Step: 2a**

Once the AWS CLI is installed, configure your access key Id and secrets key id to setup aws profile.

aws configure --profile <PROFILE\_NAME>

where PROFILE\_NAME could be a random name, this profile would be written into ~/.aws/credentials, and ~/.aws/config

AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE

AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

Default region name [None]: us-west-1

Default output format [None]: json

**Step: 3**

*Generate SSH Keys*

Execute this command for generating ssh keys

ssh-keygen -t rsa -b 4096 -N "<PASSWORD>" -C "<EMAIL\_ID>" -f "id\_rsa.< ANY\_SSHKEY\_NAME >"

Executing above command will generate two files. (i.e id\_rsa.< ANY\_SSHKEY\_NAME > and id\_rsa.< ANY\_SSHKEY\_NAME >.pub )

* Copy newly generated keys (i.e id\_rsa.< ANY\_SSHKEY\_NAME> and id\_rsa.< ANY\_SSHKEY\_NAME.pub) to the .ssh folder.
* chmod 400 ~/.ssh/id\_rsa.\*
* ssh-add ~/.ssh/id\_rsa.< ANY\_SSHKEY\_NAME >
* Enter the password given while creating the ssh keys

**Step: 4**

*Import Key Pairs created in step 3 to AWS console.*

Execute below command in your terminal

aws ec2 import-key-pair --profile PROFILE\_NAME --region us-west-1 --key-name <ANY\_SSHKEY\_NAME> --public-key-material=file://<PATH\_TO\_YOUR\_DOT\_PUB\_FILE>

* PROFILE\_NAME : profile name created in step 2a
* ANY\_SSHKEY\_NAME: name of ssh key created in step 3
* PATH\_TO\_YOUR\_DOT\_PUB\_FILE : path to id\_rsa.< ANY\_SSHKEY\_NAME>.pub file
* (Example : ~/.ssh/id\_rsa.< ANY\_SSHKEY\_NAME>.pub)

Note: I am creating my ELB, EC2 in us-west-1 region. You can choose whichever region closer to your geo location.

AWS documentation <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-available-regions>

**Step: 5**

*Create VPC and Subnets*

In this step, we are going to create VPC with name vpc-1, CIDR block for vpc will be 10.0.0.0/16, private subnets with CIDR block 10.0.2.0/24 and public subnets with CIDR block 10.0.1.0/24. Default Network ACL’s, Private Route tables with NAT gateway and Public Route tables with Internet Gateway.

**Download aws cloudformation templates from** <https://github.com/lakshmi-mahabaleshwara/drools-aws-cloudformation-templates>

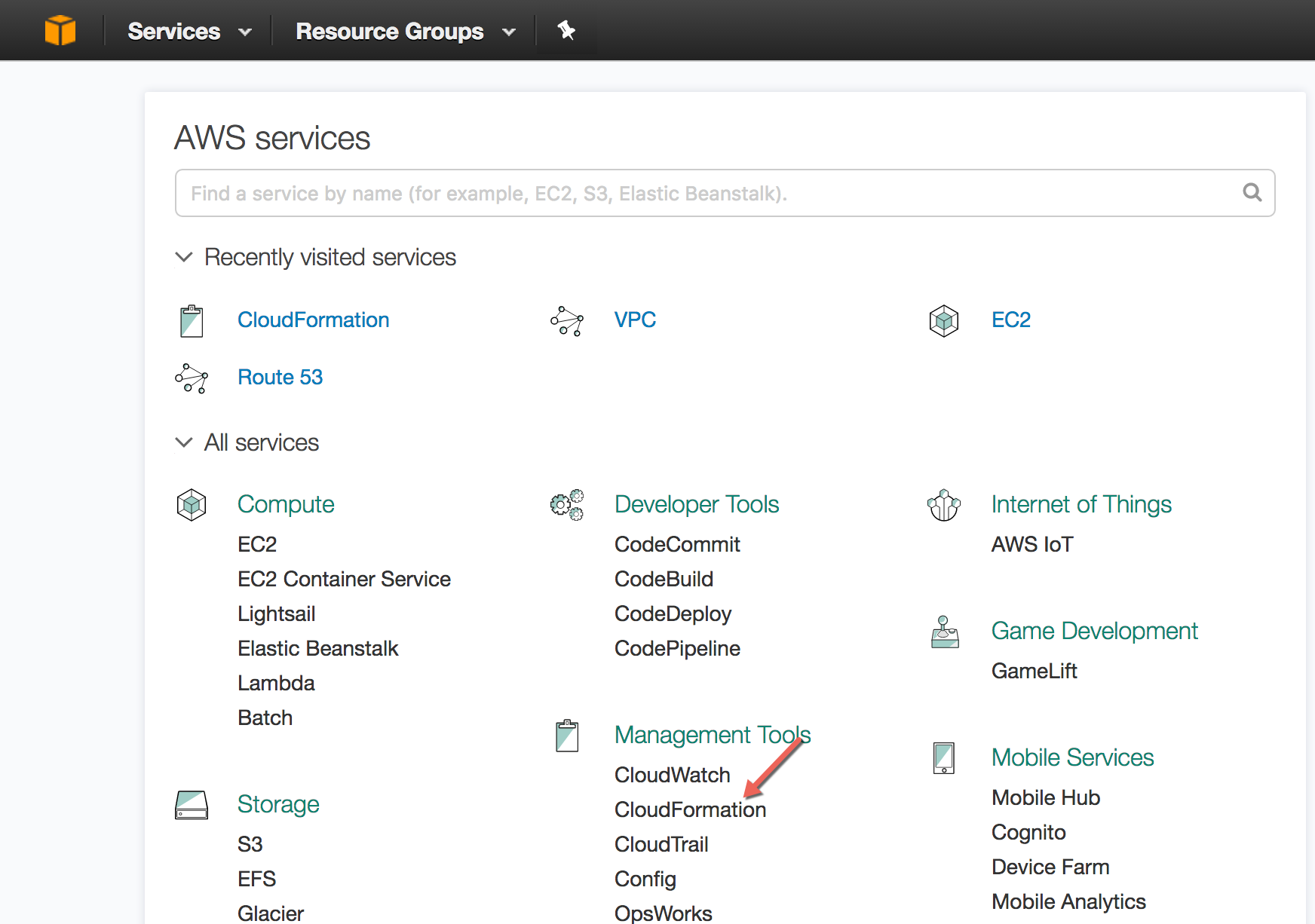
Execute

aws cloudformation create-stack --profile <PROFILE\_NAME> --region <REGION> --stack-name myproject-vpc --template-body file://</Users/xxxx/cloudformation/create\_vpc\_subnets.json> --parameters ParameterKey=subnet1AZ,ParameterValue=<AVAILIBILITY\_ZONE>

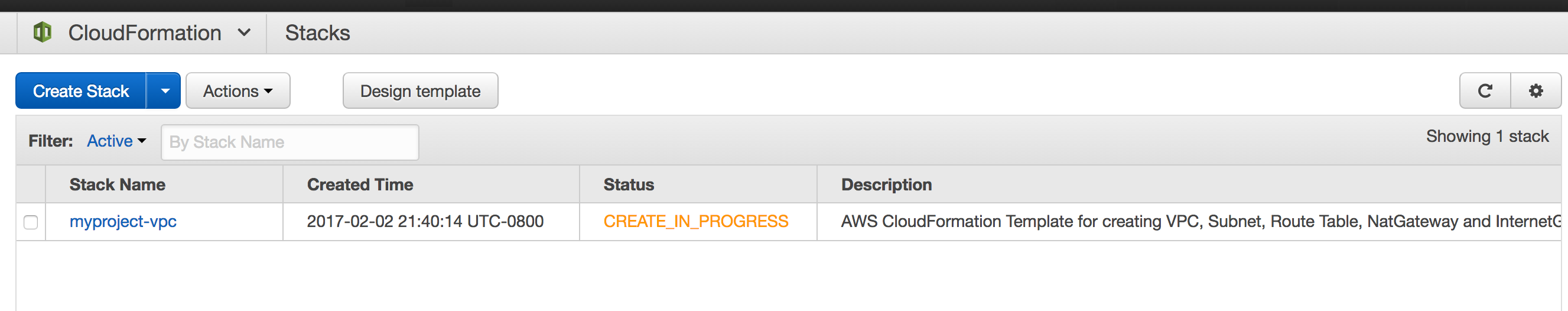
* PROFILE\_NAME : profile name created in step 2a
* REGION : closer region to your geo location, which you chose in Step 4 (I have chosen us-west-1)
* /Users/xxxx/cloudformation/create\_vpc\_subnets.json : path to create\_vpc\_subnets.json template which is downloaded from github <https://github.com/lakshmi-mahabaleshwara/drools-aws-cloudformation-templates/blob/master/aws-cloudformation/create_vpc_subnets.json>
* AVAILABILITY\_ZONE : Availability zone from your selected region (AWS Documentation <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#using-regions-availability-zones-describe> ). I have selected us-west-1a as Availability zone.

Once you execute the command, you can check the status of this cloud formation template in AWS management console.

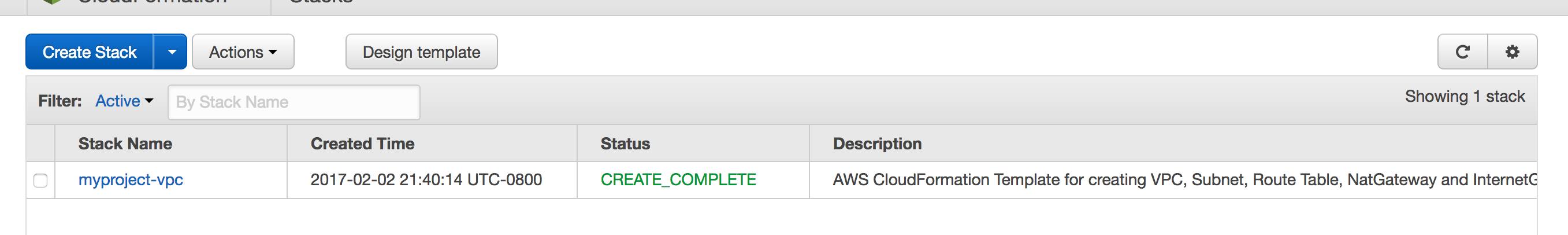
Screen :1 CloudFormation link is found under the Management tools subdivision



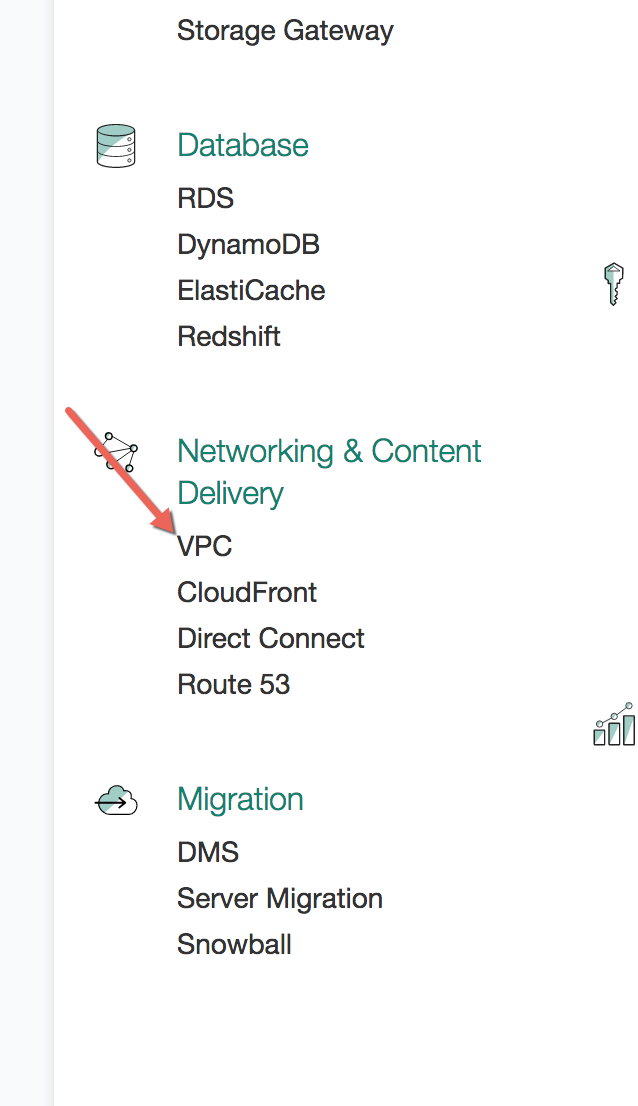
Screen:2 In progress status



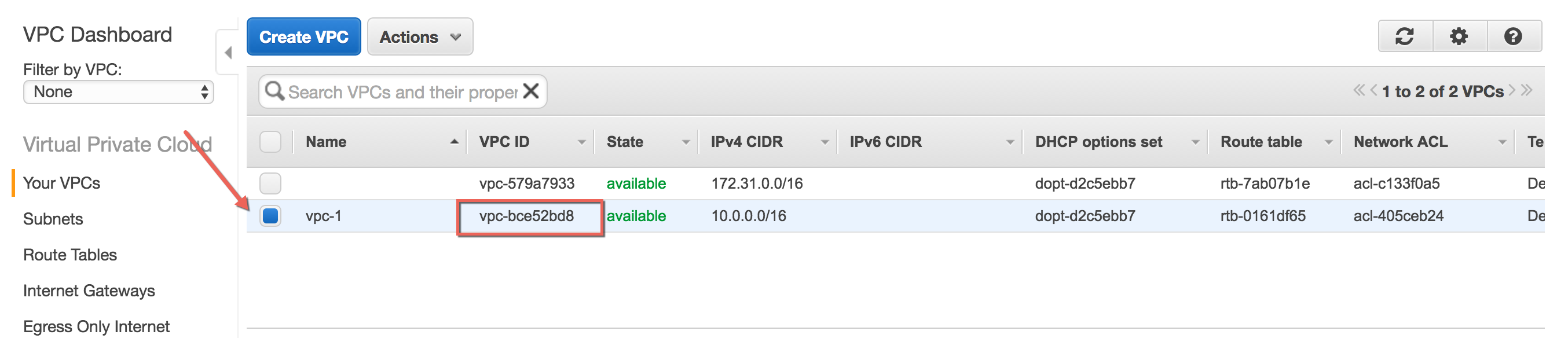
Screen:3 Once completed, VPC and subnets are created.



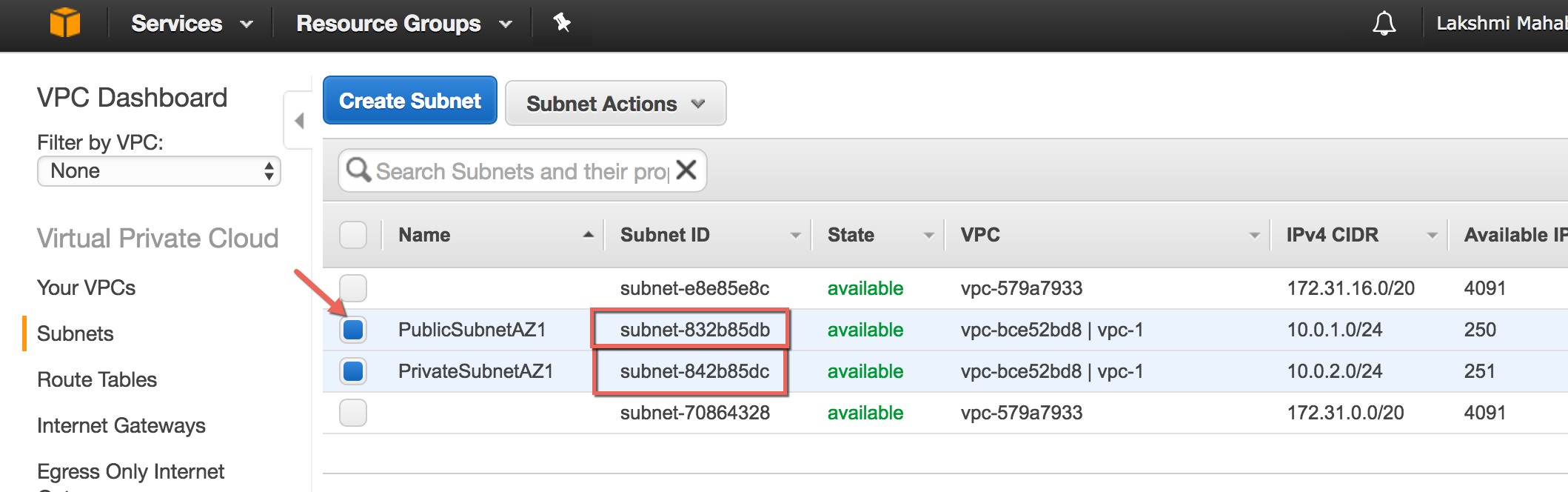
Screen:4 To Verify VPC, click VPC under Networking & Content Delivery



Screen:5 VPC



Screen: 6 Private and Public Subnets.



**Step: 6**

*Create Bastion Host*

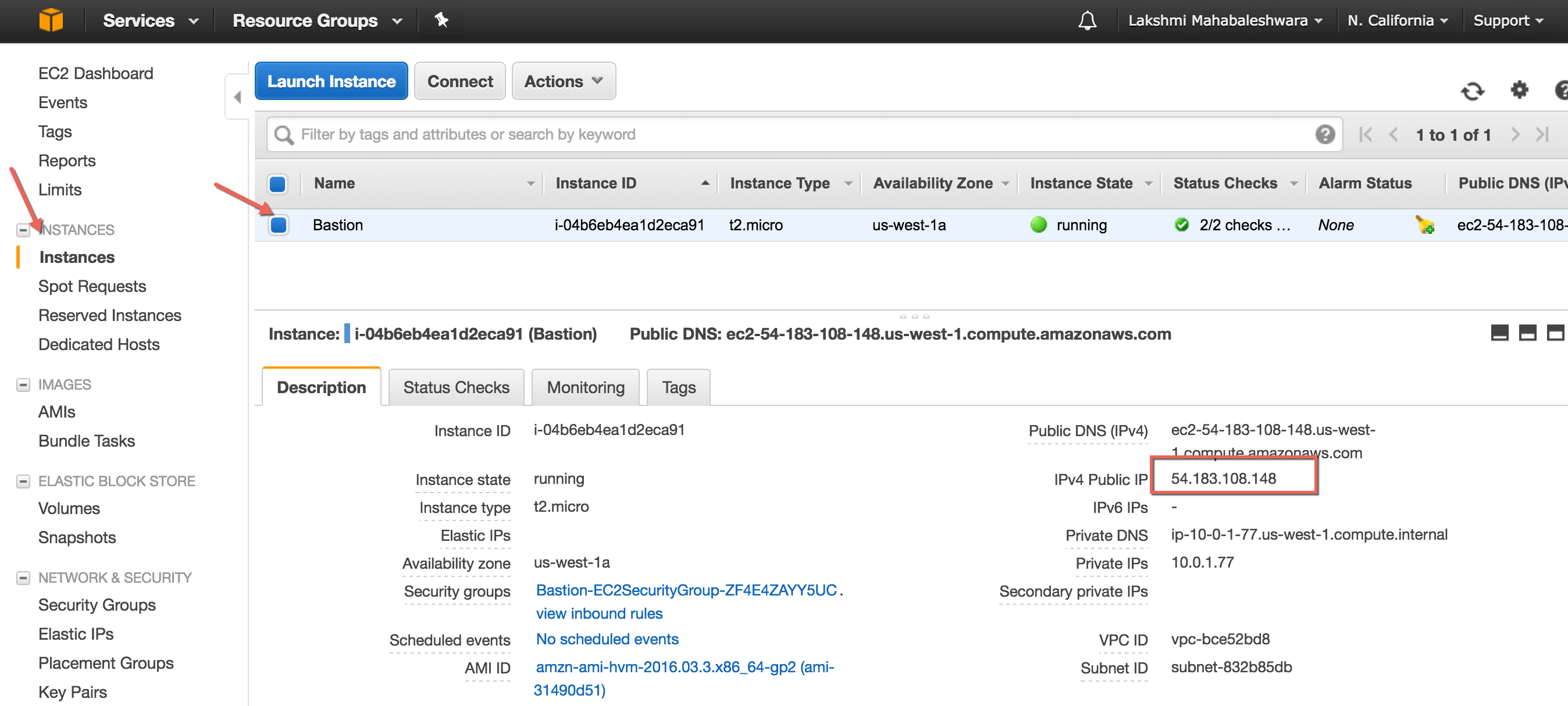
In this Step Bastion Host is created in Public subnet with ssh access to CIDR block 0.0.0.0/0 in Security Group.

aws cloudformation create-stack --profile <PROFILE\_NAME> --region <REGION> --stack-name Bastion --template-body file://</Users/xxxx/xxx/cloudformation/create\_bastion\_host.json> --parameters ParameterKey=KeyName,ParameterValue=<ANY\_SSHKEY\_NAME> ParameterKey=VpcId,ParameterValue=<VPC-ID> ParameterKey=PublicSubnetIDForBastionHost,ParameterValue=<PUBLIC\_SUBNET\_ID>

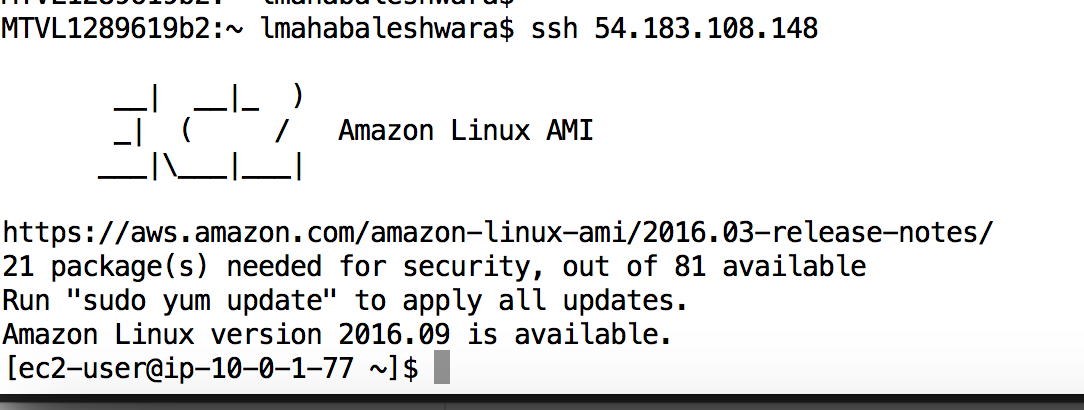
* PROFILE\_NAME : profile name created in step 2a
* REGION : closer region to your geo location, which you chose in Step 4 (I have chosen us-west-1)
* ANY\_SSHKEY\_NAME: SSH keys generated in step 3
* VPC-ID : vpc-Id from step 5, screen 5.
* PUBLIC\_SUBNET\_ID : public subnet id from step 5, screen 6.
* /Users/xxxx/xxx/cloudformation/create\_bastion\_host.json : path to create\_bastion\_host.json file downloaded from github <https://github.com/lakshmi-mahabaleshwara/drools-aws-cloudformation-templates/blob/master/aws-cloudformation/create_bastion_host.json>

Once Cloud formation template is completed, Bastion is created in EC2 instances with public IP address. Using public ip address you can ssh to the box.

Screen: 1 Bastion Instances



Screen: 2 SSH to Bastion

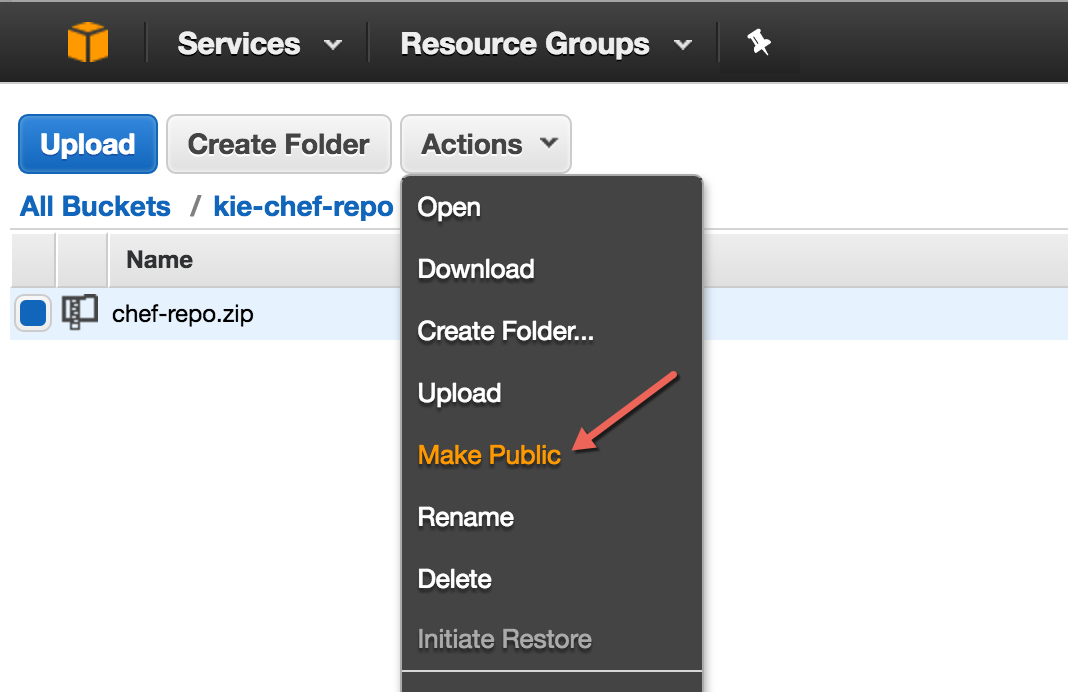


**Step: 7**

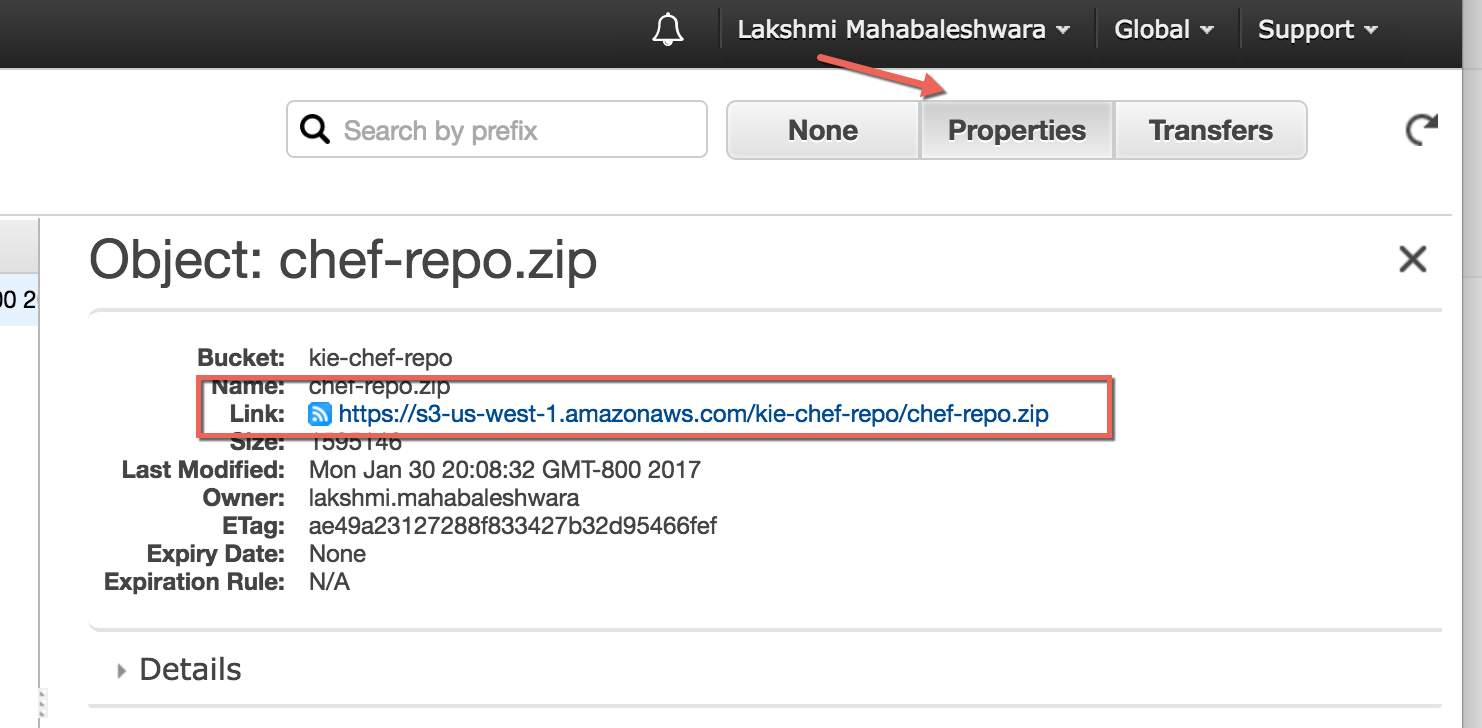
Create a Folder in S3 Bucket (AWS Documentation <http://docs.aws.amazon.com/AmazonS3/latest/UG/CreatingaFolder.html> ) to store the Chef-repo download from Github <https://github.com/lakshmi-mahabaleshwara/drools-aws-cloudformation-templates/tree/master/chef-repo>

* Run **mvn clean install** under ~/.drools-aws-cloudformation-templates/chef-repo, Once its successful it will create target folder inside drools-aws-cloudformation-templates/chef-repo/
* Upload chef-repo.zip generated under target folder to S3 bucket newly created folder. Once it’s uploaded, select the folder using check box and make it public under Actions drop down. (Screen 1)
* Now click on Properties button in right side and copy the link for this s3 bucket. (Screen 2)
* Paste the copied s3 link in code <https://github.com/lakshmi-mahabaleshwara/drools-aws-cloudformation-templates/blob/master/aws-cloudformation/create_ec2_private_chef.json#L317> this is will download the chef-repo to ec2 instances from s3 bucket.

Screen: 1 Making s3 bucket public



Screen: 2 S3 link



**Step: 8**

*Create Private Subnet*

This template creates private subnet, with ELB, two EC2 instances behind ELB, installs tomcat, Kie jars and wars through chef-solo.

Execute this command

aws cloudformation create-stack --profile <PROFILE\_NAME> --region <REGION> --stack-name myPrivateEC2 --template-body file://</Users/xxxx/xxx/cloudformation/create\_ec2\_private\_chef.json --parameters ParameterKey=KeyName,ParameterValue=<ANY\_SSHKEY\_NAME> ParameterKey=VpcId,ParameterValue=<VPC-ID> ParameterKey=PrivateSubnetIds,ParameterValue=<PRIVATE\_SUBNET\_ID>

ParameterKey=PublicSubnetIds,ParameterValue=<PUBLIC\_SUBNET\_ID>

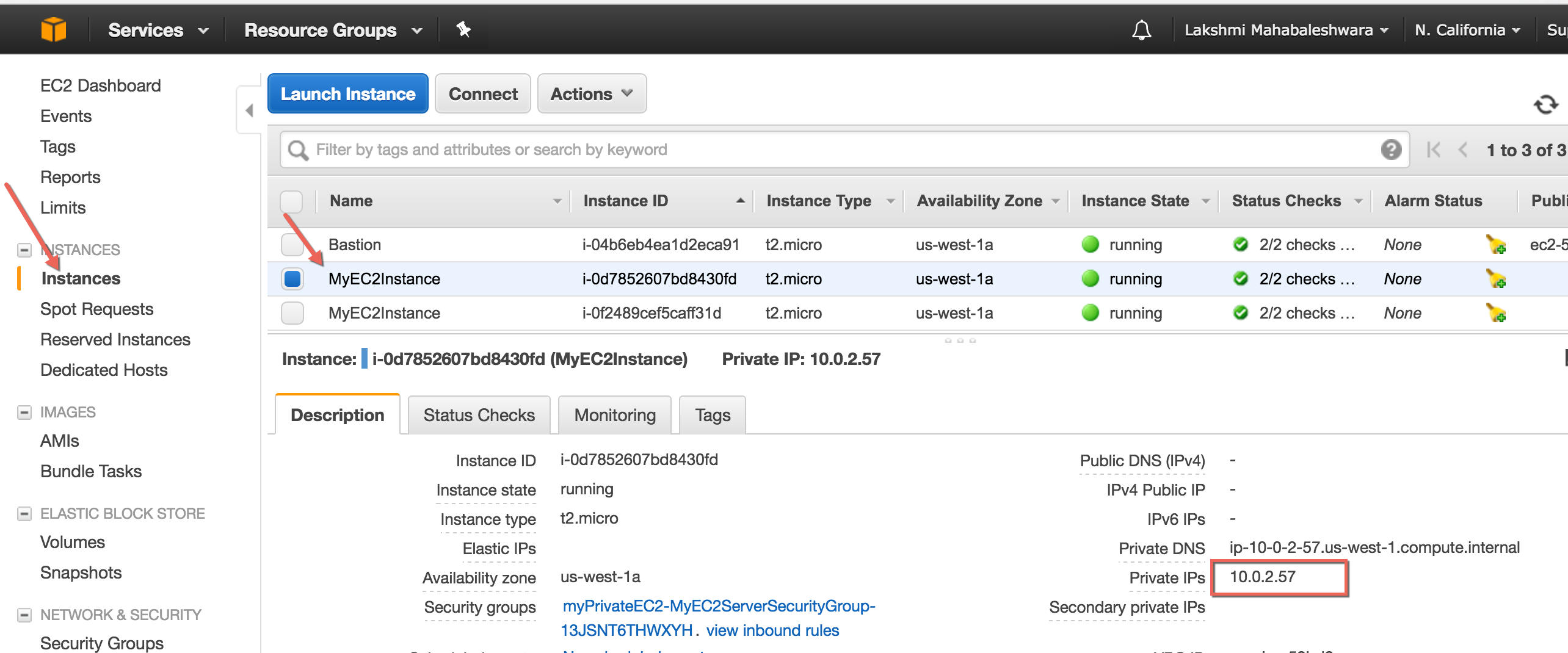
* PROFILE\_NAME : profile name created in step 2a
* REGION : closer region to your geo location, which you chose in Step 4 (I have chosen us-west-1)
* ANY\_SSHKEY\_NAME: SSH keys generated in step 3
* VPC-ID : vpc-Id from step 5, screen 5.
* PUBLIC\_SUBNET\_ID : public subnet id from step 5, screen 6.
* PRIVATE\_SUBNET\_ID : private subnet id from step 5, screen 6.
* / Users/xxxx/xxx/cloudformation/create\_ec2\_private\_chef.json: path to create\_ec2\_private\_chef.json file downloaded from github <https://github.com/lakshmi-mahabaleshwara/drools-aws-cloudformation-templates/blob/master/aws-cloudformation/create_ec2_private_chef.json>

Once created, you can ssh to private instances after ssh to Bastion.

Two EC2 instances with Autoscaling group is created.

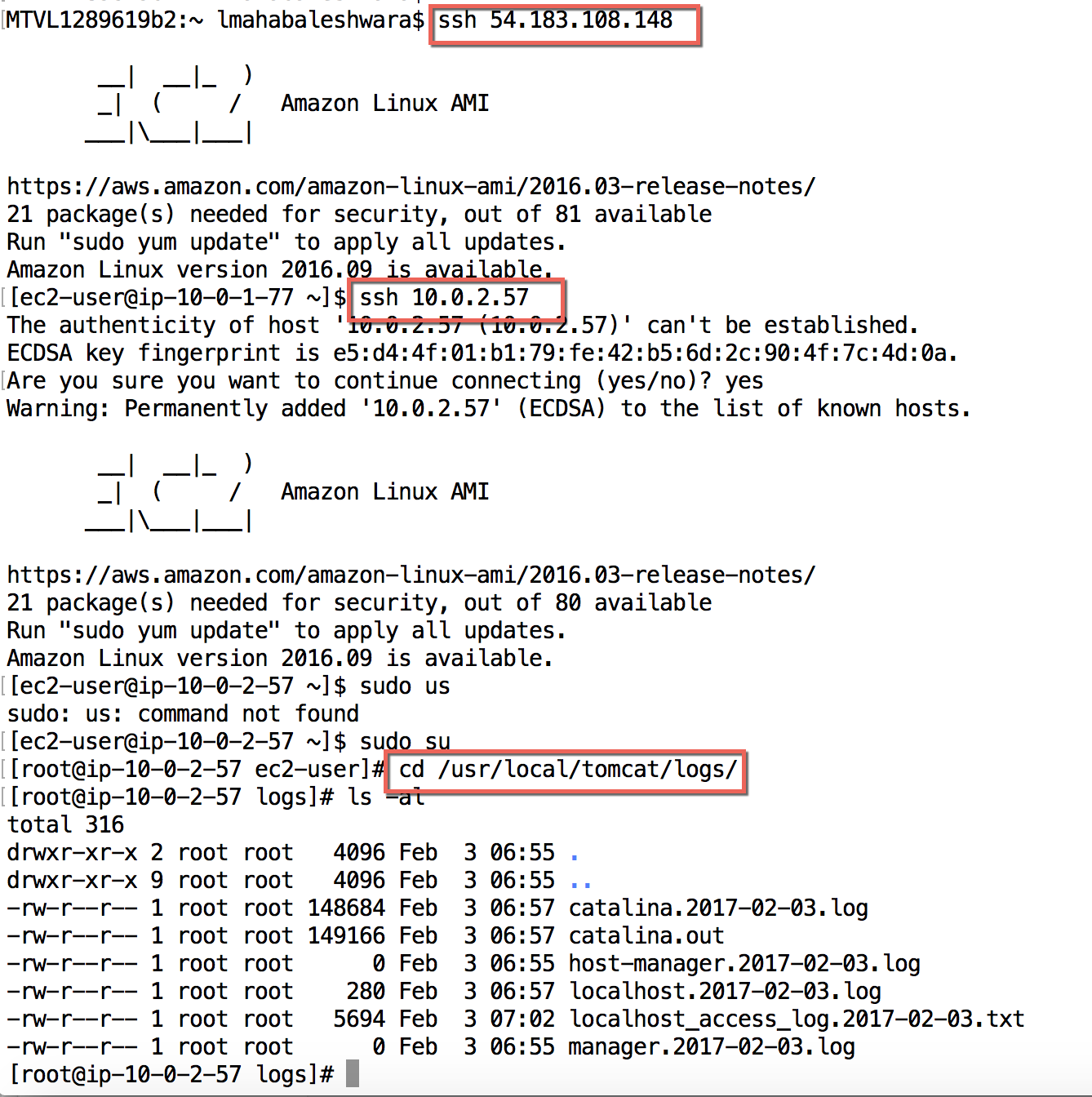
Tomcat logs are located at usr/local/tomcat/logs

Screen:1 EC2 instances



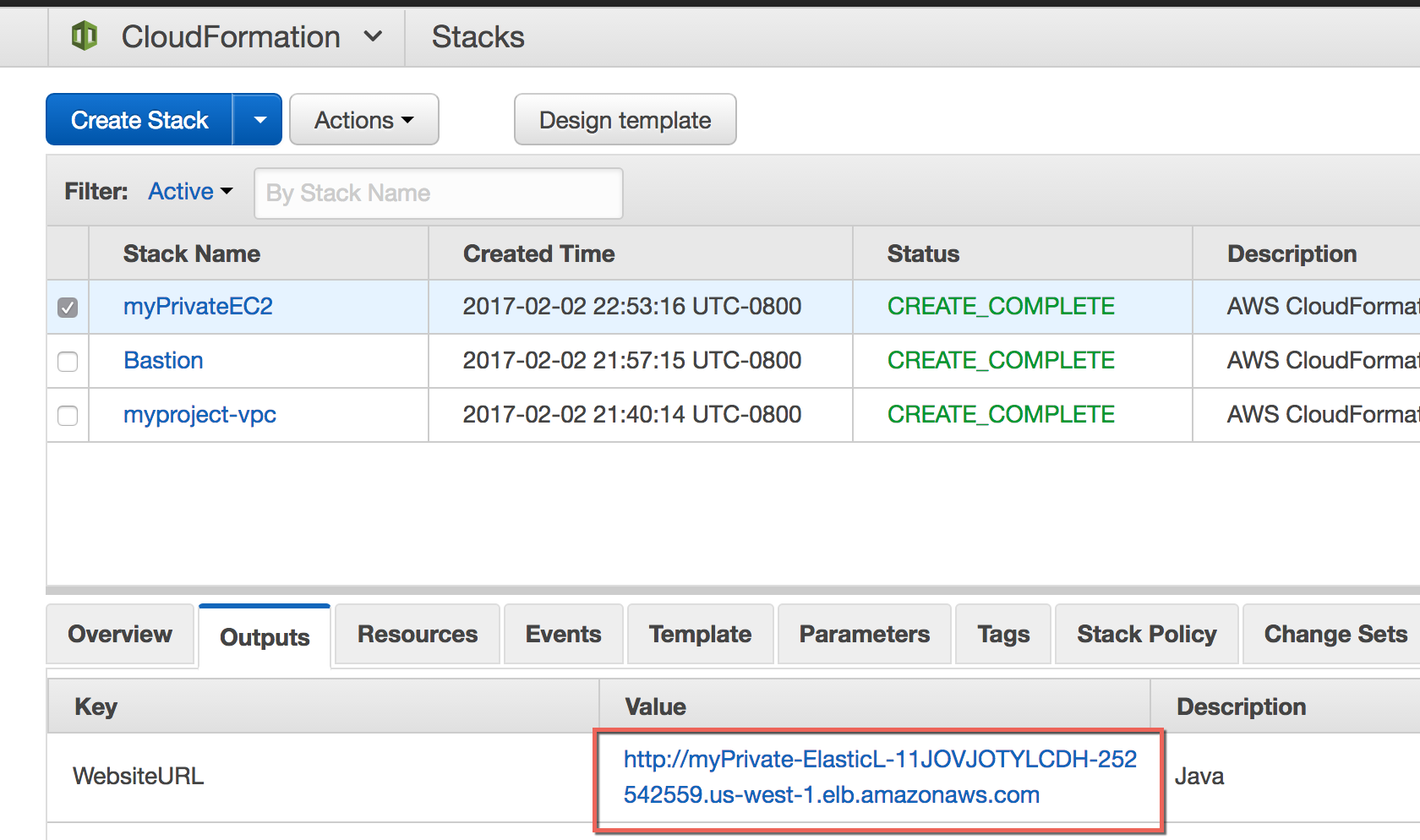
Screen:2 ssh to private EC2 instances

Get the IP address of private instance from previous step and ip address of Bastion from step 6



Finally, We are all set to launch Kie-Workbench application in Browser.

Screen: 3 Get DNS name of ELB to launch in Browser from Cloudformation



Screen: 4 Launch Kie-wb url

I have used username/password as workbench/workbench1! In tomcat-users.xml. <https://github.com/lakshmi-mahabaleshwara/drools-aws-cloudformation-templates/blob/master/chef-repo/cookbooks/application/templates/default/tomcat-users.xml.erb#L41>

URL : <http://myprivate-elasticl-11jovjotylcdh-252542559.us-west-1.elb.amazonaws.com:8080/kie-wb>

