SCALING DEVOPS DEPLOYMENTS WITH AWS CODE PIPELINE, BLAZEMETER, & DYNATRACE APPMON

CMG Boston – April 2017

Joe Sicree

Sr. Consultant @ CGI

@JoeSicree. -- Joe.Sicree@cgi.com

CGI

Rob Jahn

Technical Director @ Elyxor

Rob.Jahn@elyxor.com

elyxor

SUMMARY

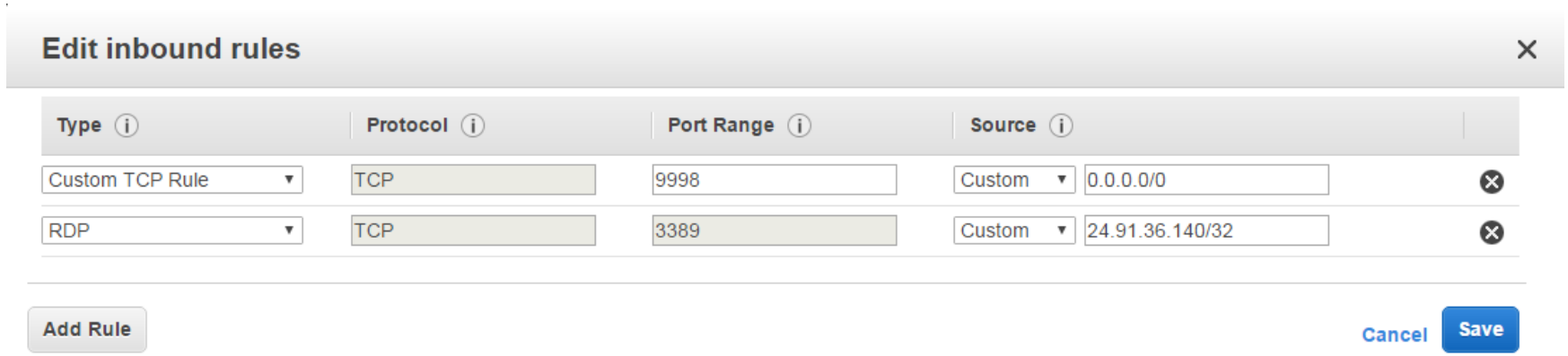
1. Setup Dynatrace AppMon on EC2
   - EC2 Instance & Inbound Rules
   - Install DT
   - Configure Windows Firewall
2. Deploy DemoApp on EBS
   - EBS Domain
   - S3 Bucket for DT agent installer
   - Customize EBS Extensions
   - Validate EBS App and Agent install
3. Verify Setup
   - Configure Demo Application Profile
   - Make requests and verify PurePath
4. AWS Pipeline Configuration

STEP 1

SETUP DYNATRACE APPMON ON EC2

EC2 INSTANCE & INBOUND RULES

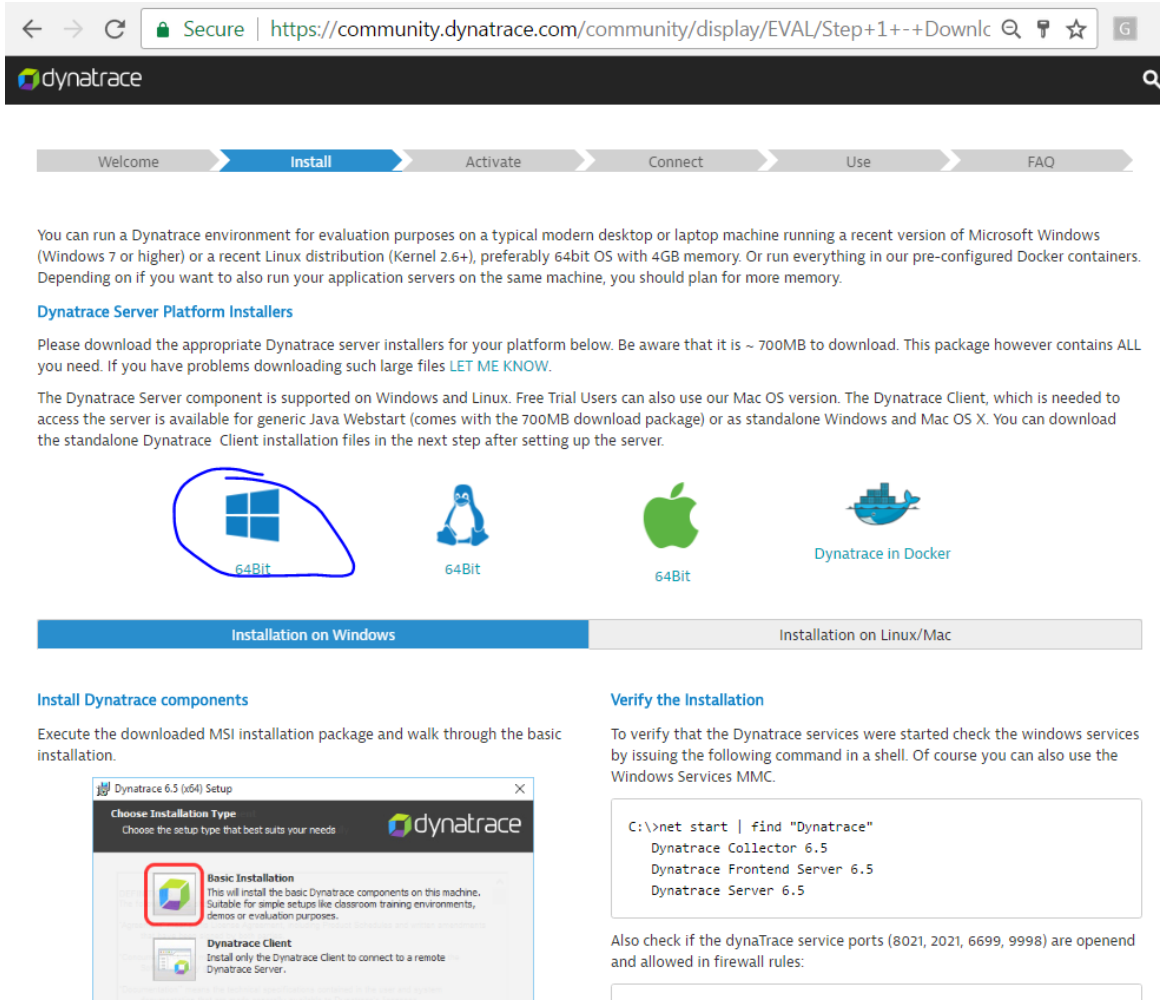
1. Launch 'Microsoft Windows Server 2016 Base" AMI as "t2.large" as to have 8GB memory. Dynatrace AppMon only runs on Windows. The default of 30GB is enough for R&D purposes.
2. For Inbound rules, need to add RDP and port for Dynatrace agent. You can adjust source, but I made RDP my IP only for Demo



3. You need to create or re-use a security Key Pair as to RDP in. I called mine (PipelineDemo) and used same one for Elastic Bean Stalk (EBS).

INSTALL DT

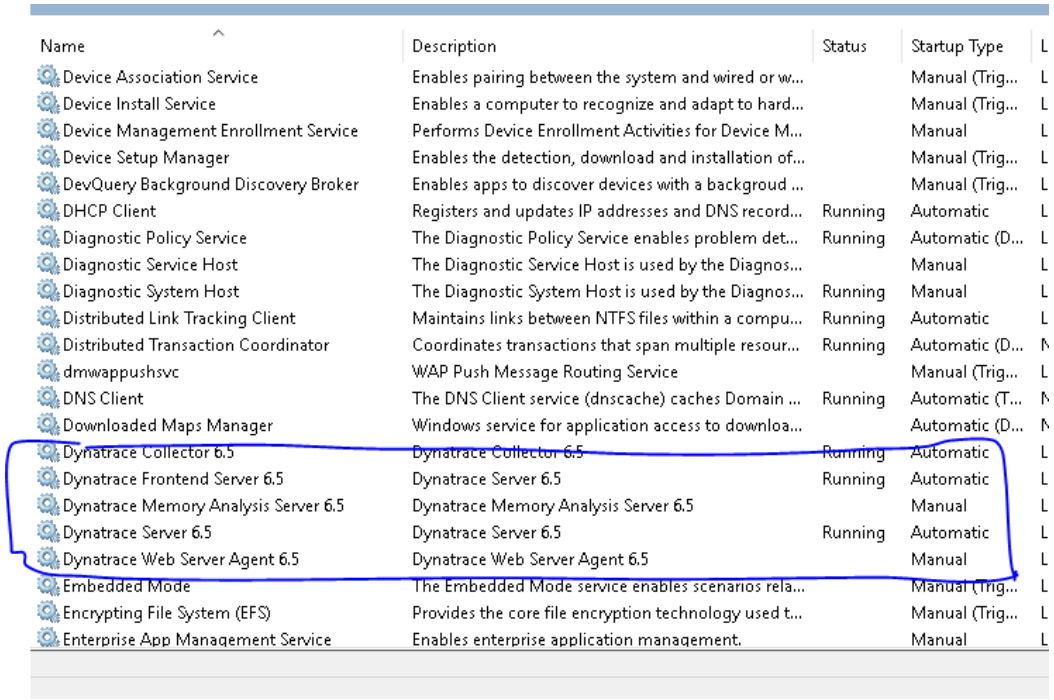
1. RDP into EC2 instance.
2. Register and download DT AppMon installer - http://bit.ly/dtpersonal
   Once verified you can link to the download page (see image to the right)
3. You will be emailed a license Key. Save this to local drive on the EC2 instance.
4. Download 64Bit Windows and install basic installation. Choose yes to install windows services

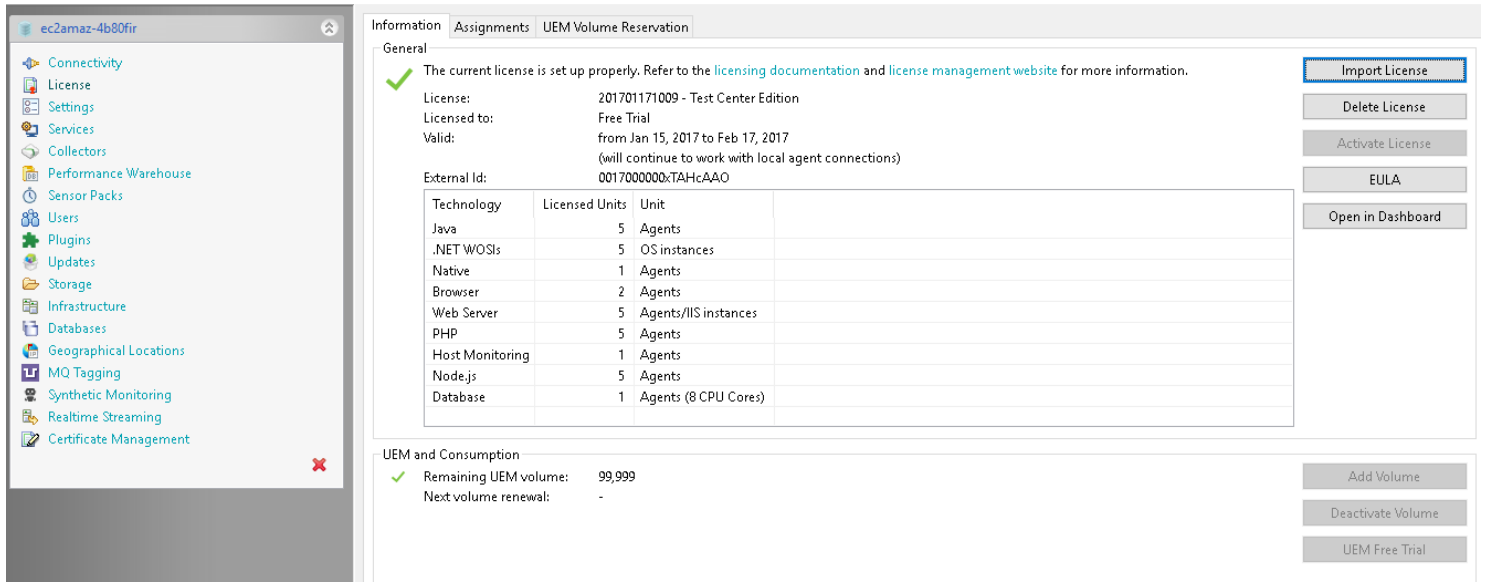


INSTALL DT

1. DT should now be running as service (see right image)
2. Click on AppMon Icon.

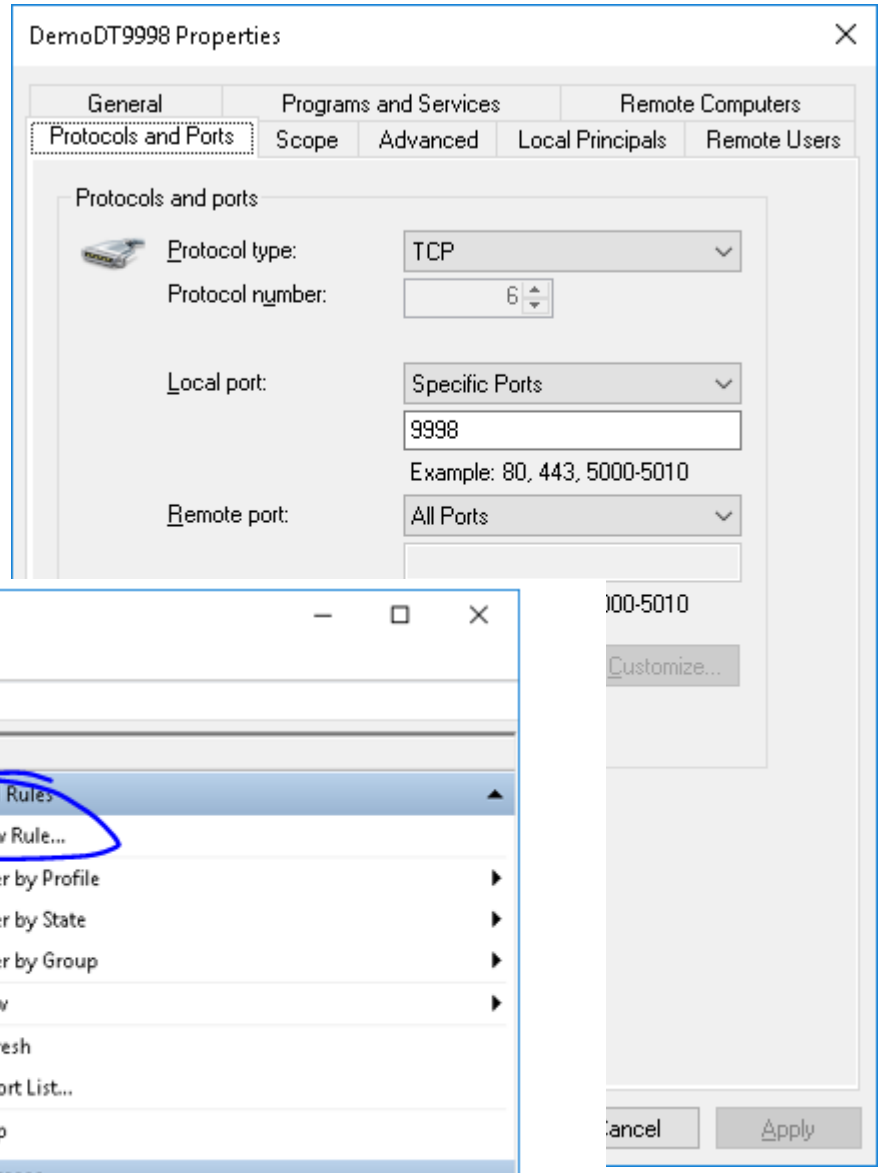
3. From the Settings→Dynatrace Server option goto license. Then import license from file

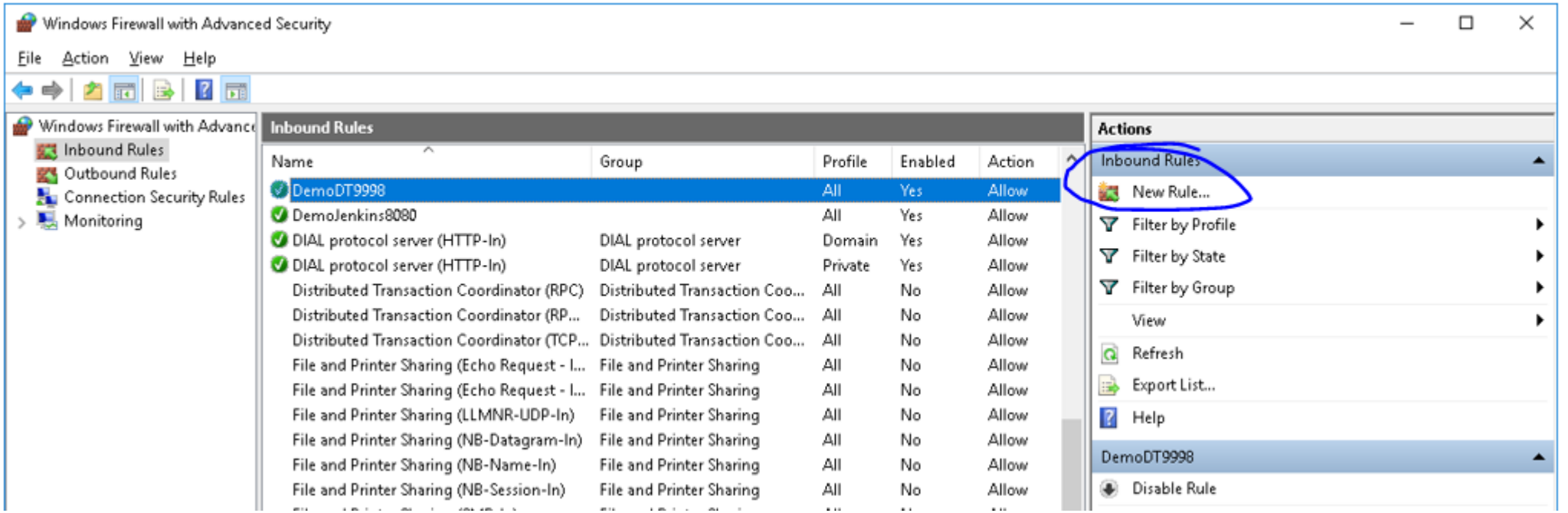




CONFIGURE WINDOWS FIREWALL

In addition to EC2 inbound rule, configure Windows firewall to allow Agents to communicate. This is because EC2 is locked down by default.



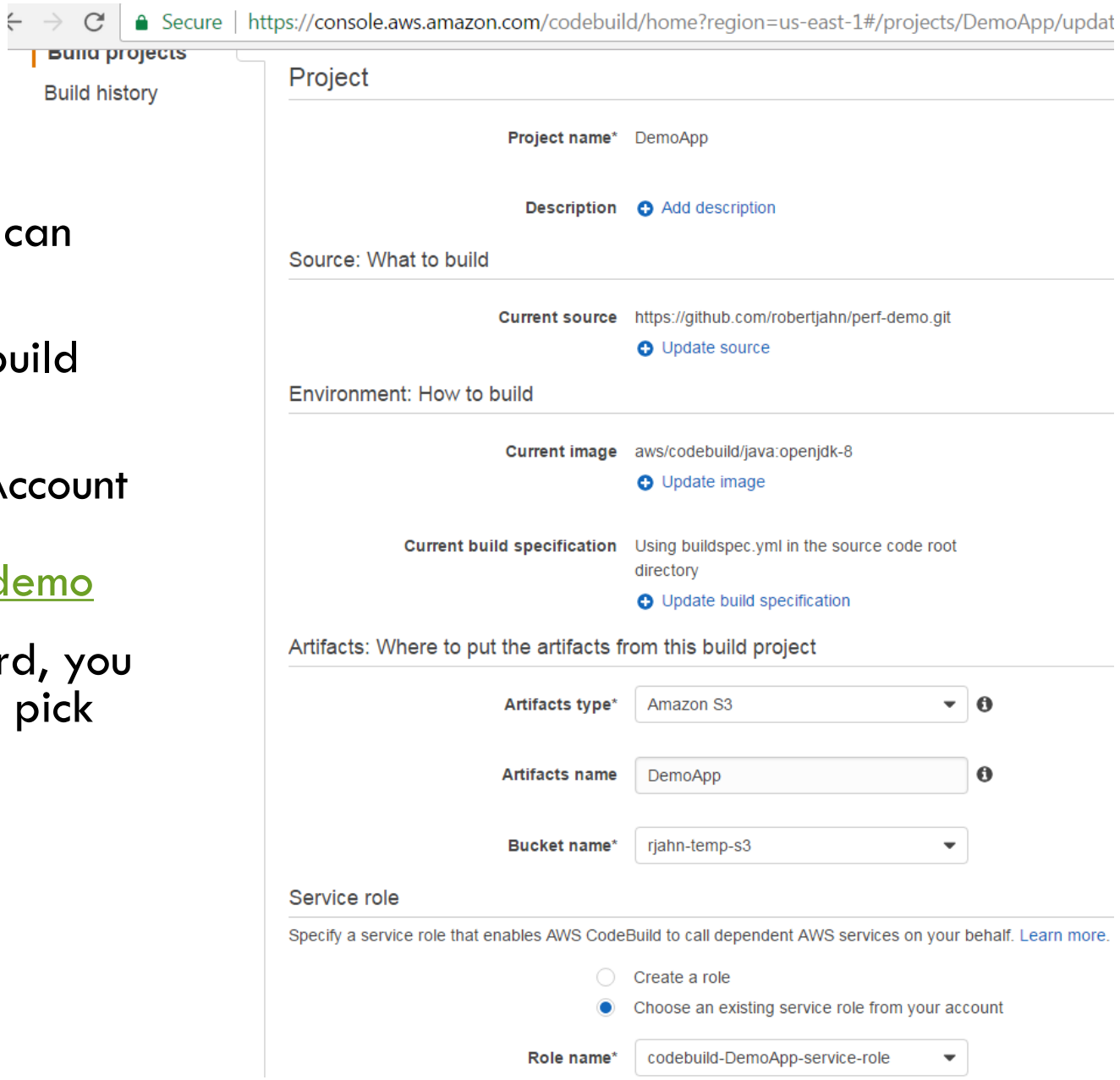


STEP 2

DEPLOY DEMOAPP ON EBS

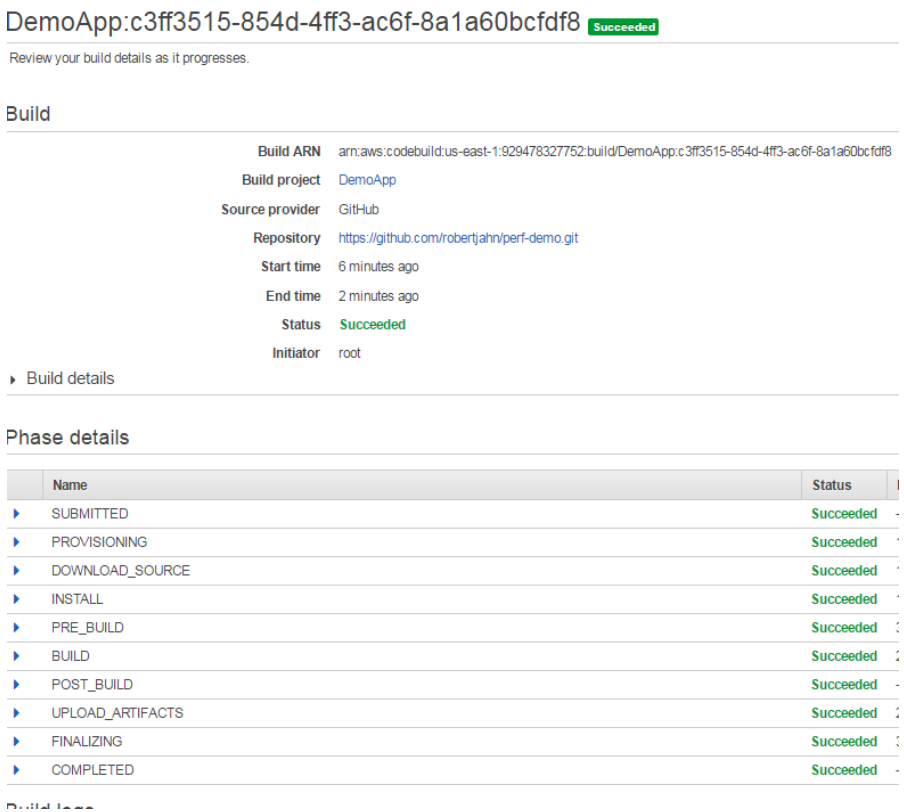
CODE BUILD

1. We need to make ZIP so that we can make the EBS environment.
2. You can do this with AWS Code build and getting code from GitHub.
3. You will need to have a GitHub Account and make a fork from https://github.com/jsicree/perf-demo
4. In the CodeBuild New App Wizard, you will enter GitHub credentials and pick the repo
5. To the right is my Demo setup.
6. Build file will goto

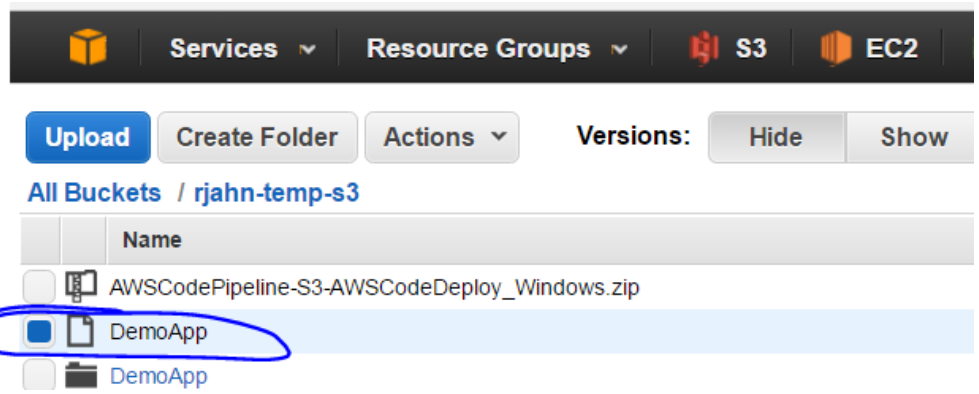


CODE BUILD

1. Successful build will look like this



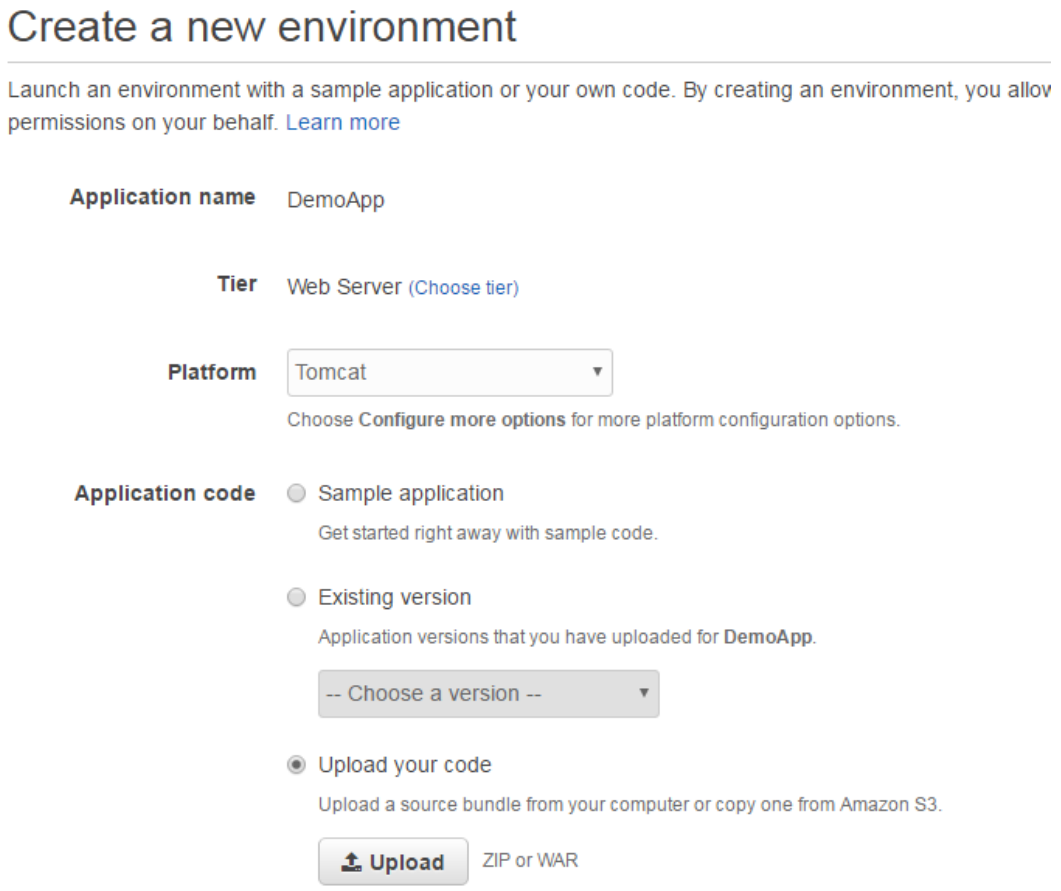
2. The ZIP artifact will go to S3 that you defined (for me this was rjahn-temp-s3).



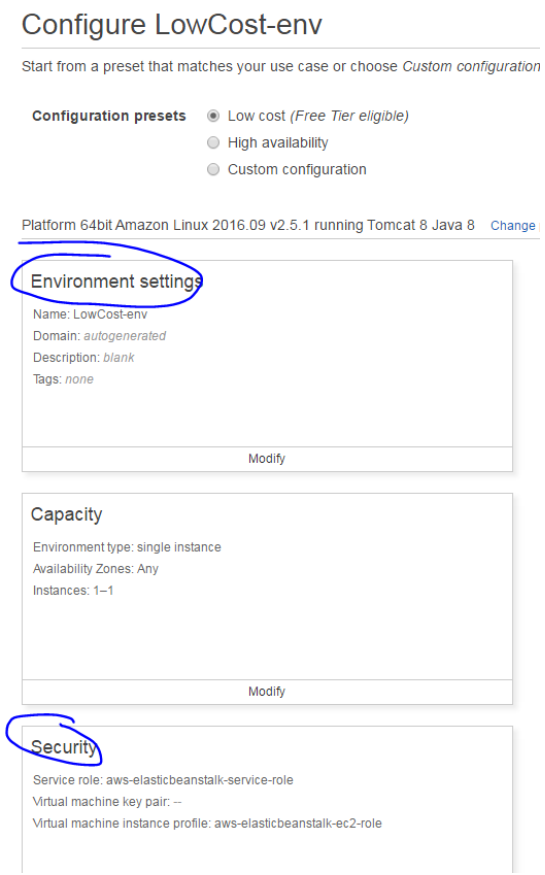
3. However, EBS cannot connect to S3, so you need to download ZIP to local drive and then later pick this file.

EBS DOMAIN

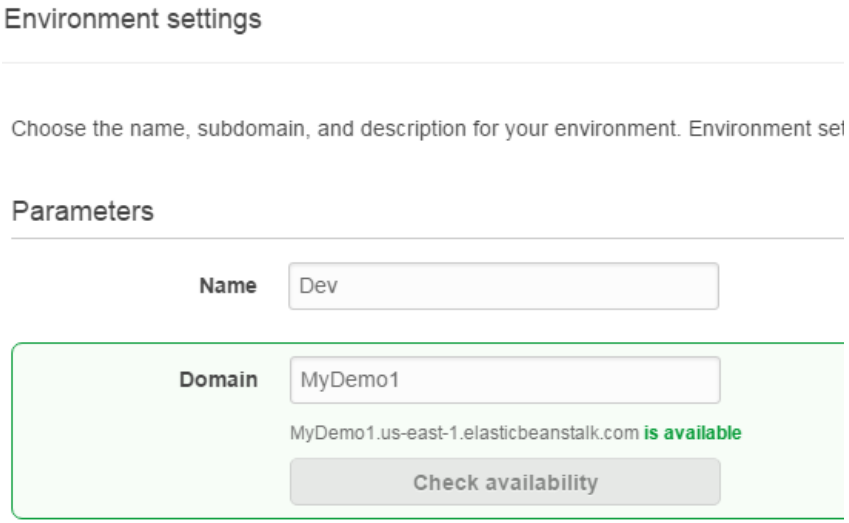
1. Created based “web server environment. Choose preconfigured “Tomcat”
2. Upload the ZIP file you created in the CodeBuild Step

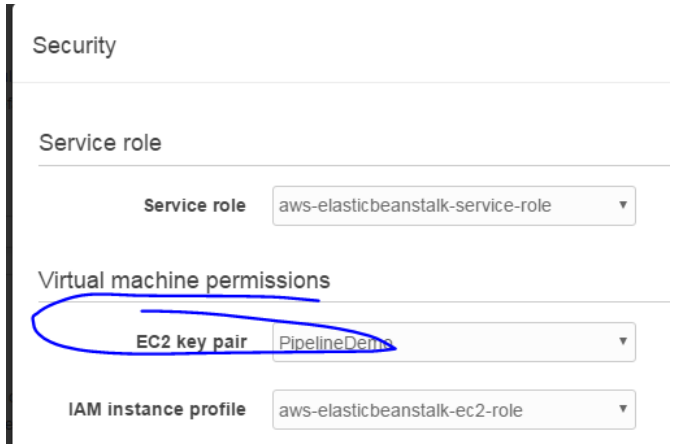


EBS DOMAIN



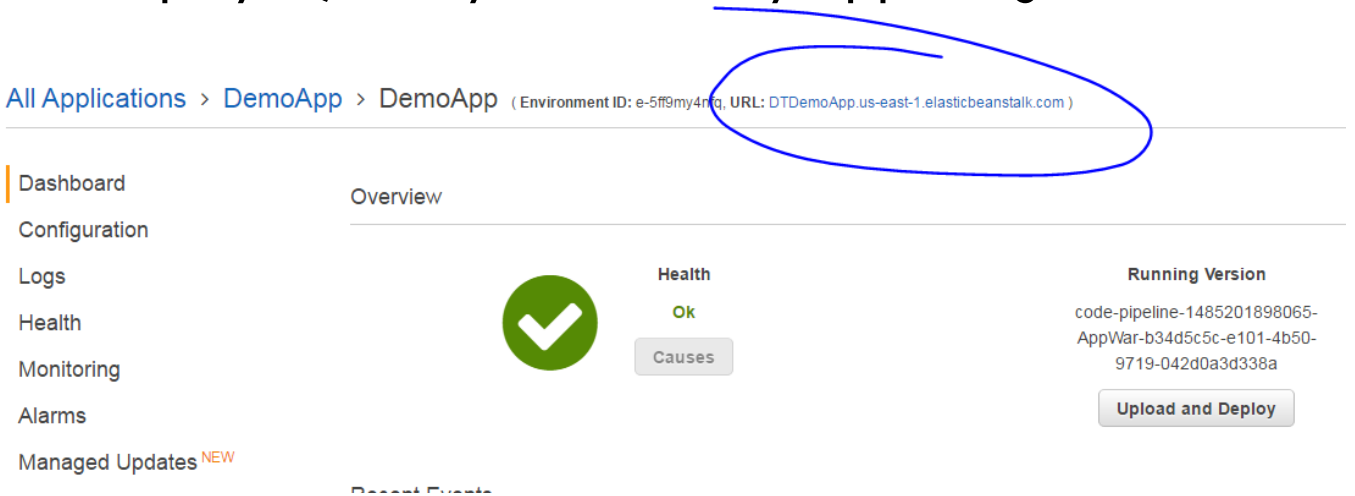
1. I choose “free tier” (which will be a t1.micro)
2. In environment, adjust the Name and Domain. Domain will give you your own URL.
3. In security, configure to use your KeyPair. I used same as DT Server (PipelineDemo) for R&D.



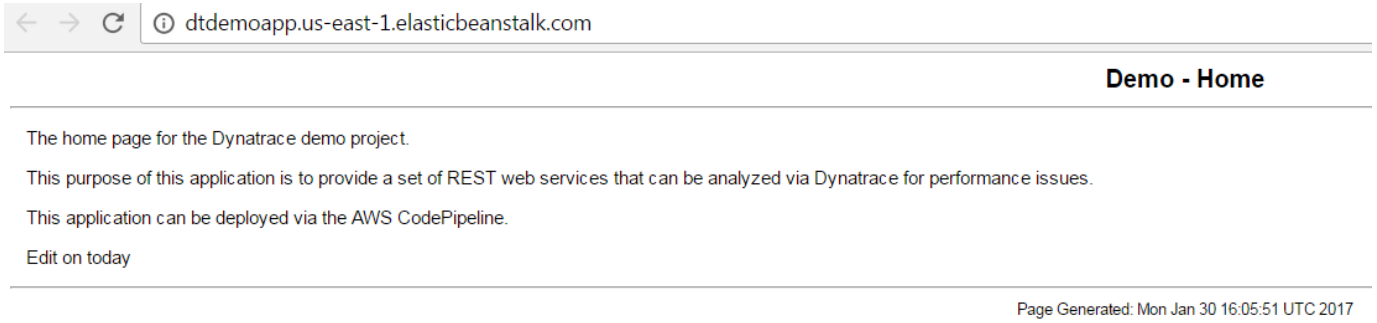


EBS DOMAIN

1. Once deployed, then you can verify app using the EBS Domain URL.

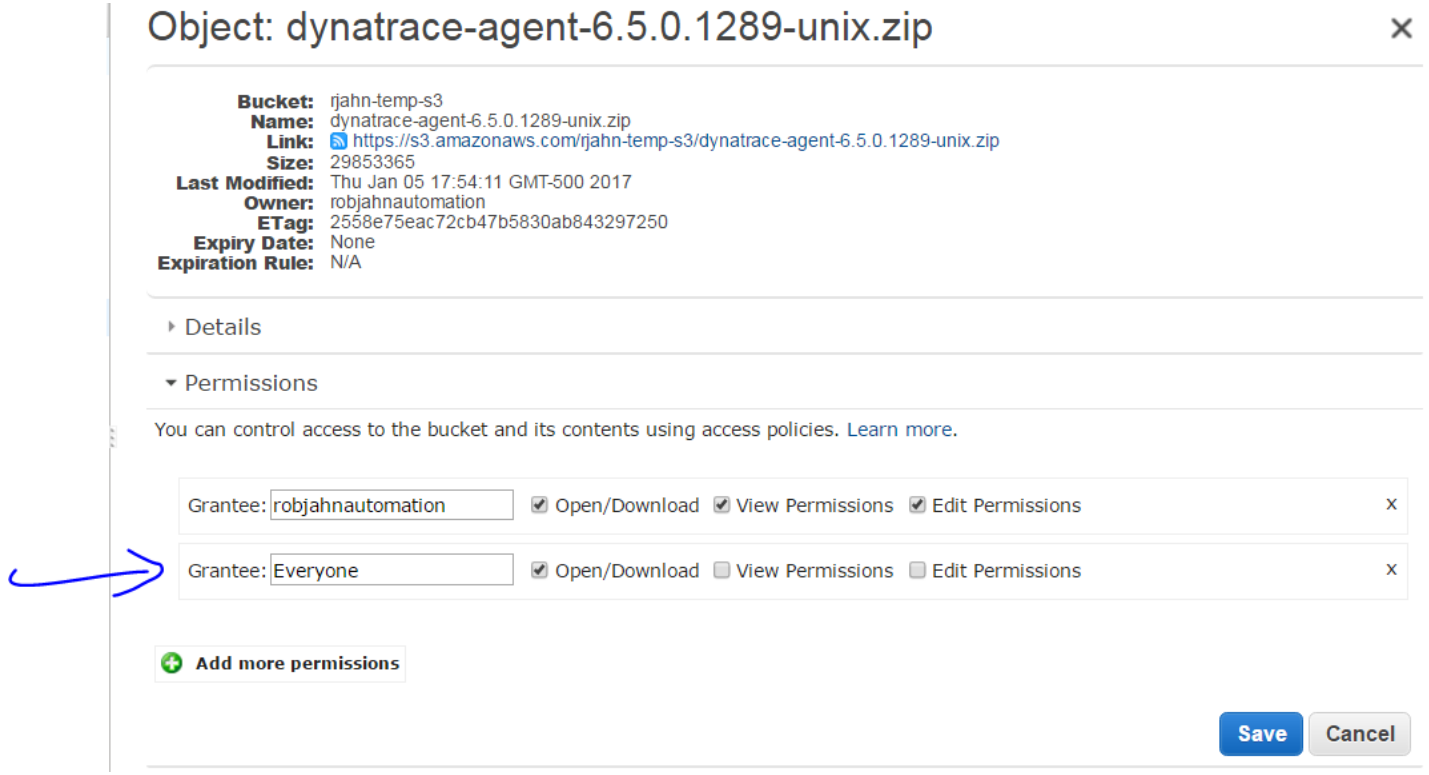


2. App should show App home page in Browser



S3 BUCKET FOR DT AGENT INSTALLER

1. You can get agent download from DT site
   - https://community.dynatrace.com/community/display/DOCDT65/Install+Agents
2. For demo I was using "dynatrace-agent-6.5.0.1289-unix.zip"
3. Copy this ZIP to you S3 Bucket. During the deployment, this file will be copied so you need to open up S3 permissions.
4. You will need the "Link" to file in EBS Extensions



CUSTOMIZE EBS EXTENSIONS

- Extensions are way to “boot strap” pre, during, and post installation actions. See AWS docs for overview

  http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/customize-containers-ec2.html

- EBS Extensions scripts are in the build. See this location of the “webapp”

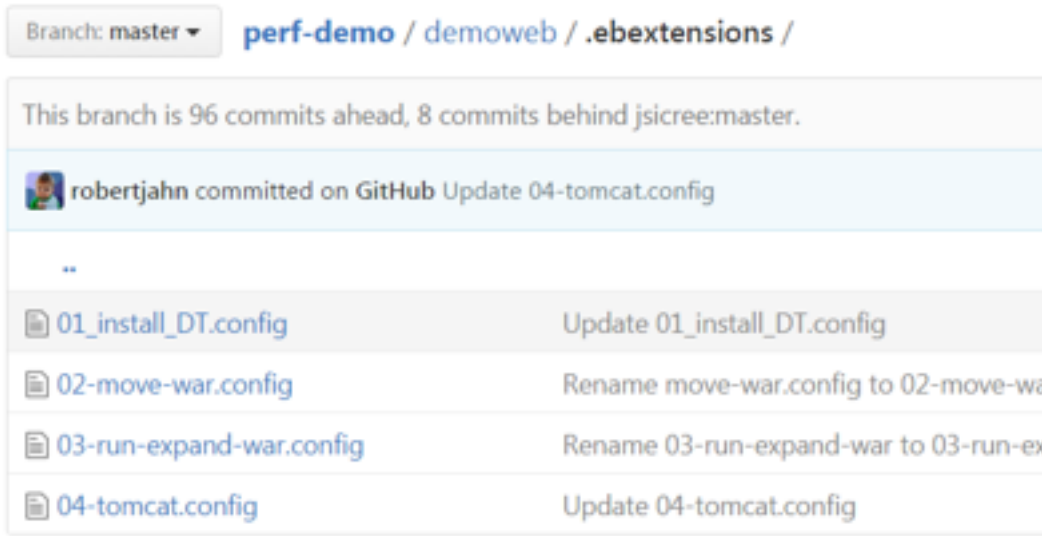
  https://github.com/jsicree/perf-demo/tree/master/demoweb/.ebextensions

- EBS Extensions scripts processed in alphabetical order. For demo we broke logic into multiple files, but they could be combined.

- We are using EBS Extensions -to do a few things
  - **01_install_DT.config** - Downloads the Dynatrace agent from an S3 bucket and installs the agent.
  - **02-move-war.config** - Creates a shell script called 00expand-war.sh that will expand any war file deployed to Tomcat.
  - **03-run-expand-war.config** - Executes the shell script 00expand-war.sh.
  - **04-tomcat.config** - Configures Tomcat with the Dynatrace agent. EBS Extensions scripts are in the build. See this location of the “webapp”

EBS CONFIGURATION

To get DT agent on EBS, need to use EB extensions as part of the deploy to install agent and to configure the JVM arguments.



https://github.com/robertjahn/perf-demo/blob/master/demoweb/.ebextensions/01_install_DT.config

28 lines (25 sloc) | 863 Bytes

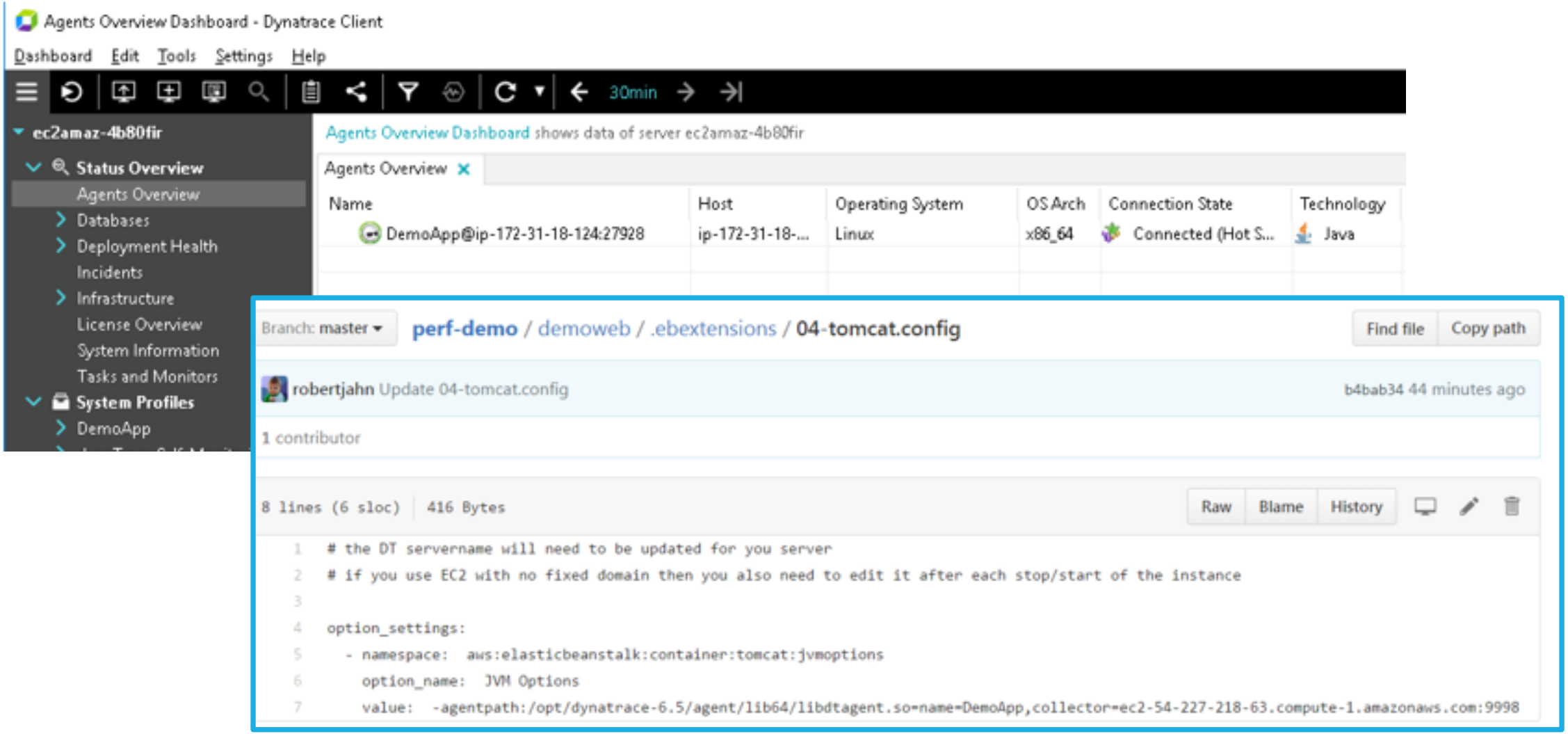
```
files:
  "/tmp/dynatrace-agent-6.5.0.1289-unix.zip":
    mode: "000755"
    owner: root
    group: root
    source: https://s3.amazonaws.com/rjahn-temp-s3/dynatrace-agent-6.5.0.1289-unix.zi

  "/opt/elasticbeanstalk/hooks/appdeploy/pre/99_install_dynatrace.sh":
    mode: "000755"
    owner: root
    group: root
    content: |
          #!/bin/bash
          if [ ! -d /opt/dynatrace-6.5 ]
          then
          #unzip_dynatrace
          sudo unzip -o /tmp/dynatrace-agent-6.5.0.1289-unix.zip -d /tmp
          #install_dynatrace in silent mode
          cd /opt
          sudo java -jar /tmp/dynatrace-agent-6.5.0.1289-unix.jar -y
          fi

commands:
   install_dynatrace:
      command: /opt/elasticbeanstalk/hooks/appdeploy/pre/99_install_dynatrace.sh
      cwd: /opt
      test: "[ -d /opt/dynatrace-6.5 ]"
```

EBS CONFIGURATION

DT agent coming in from EBS



CUSTOMIZE EBS EXTENSIONS

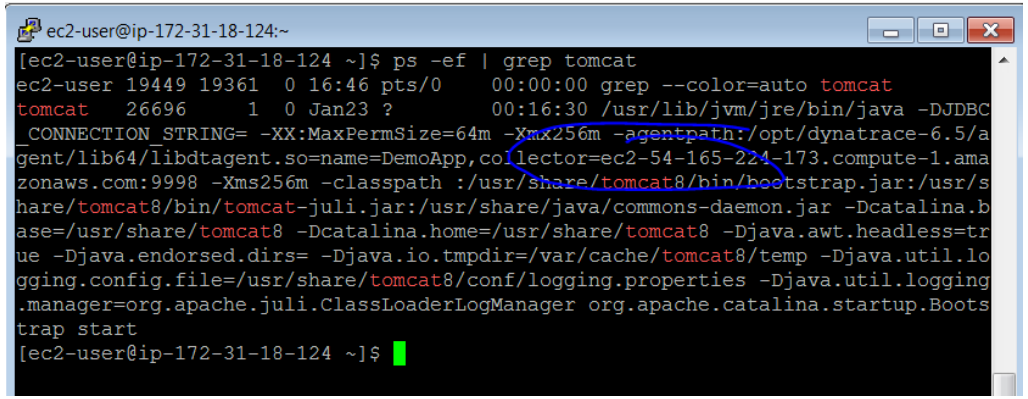
- You need to adjust "**01_install_DT.config**" with your s3 bucket location. Get URL from the s3 file properties

- You need to adjust "**04-tomcat.config**" with location for your DT Collector. Get Public URL from the EC2 console page, server properties

VALIDATE EBS APP AND AGENT INSTALL

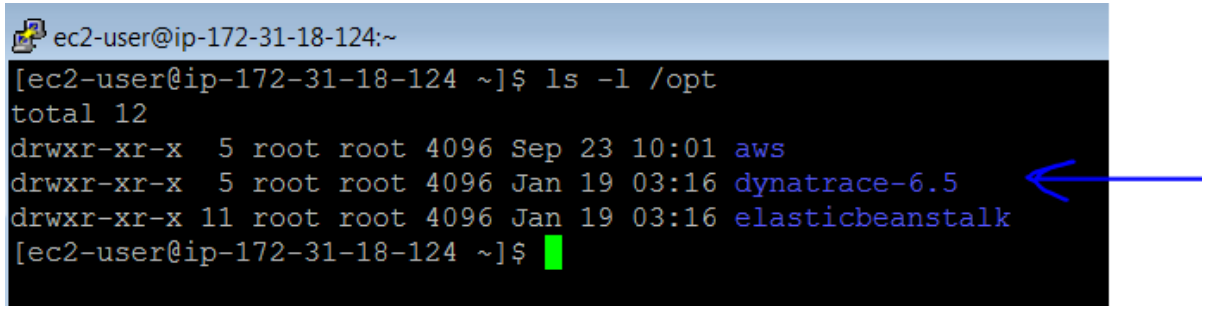
- Once you adjust these configurations, you need to do another build and deploy.
- You can verify the scripts worked in the following manner

1. SSH (I use Putty) to the App Linux instance.
2. Verify demoapp ZIP is expanded (NOTE: only in code pipeline was this an issued. The EBS Code deploy expands automatically)
   - You should see expanded zip here: /var/lib/tomcat8/webapps/ROOT
3. See if tomcat options have adjusted controller
   - ps –ef | grep tomcat

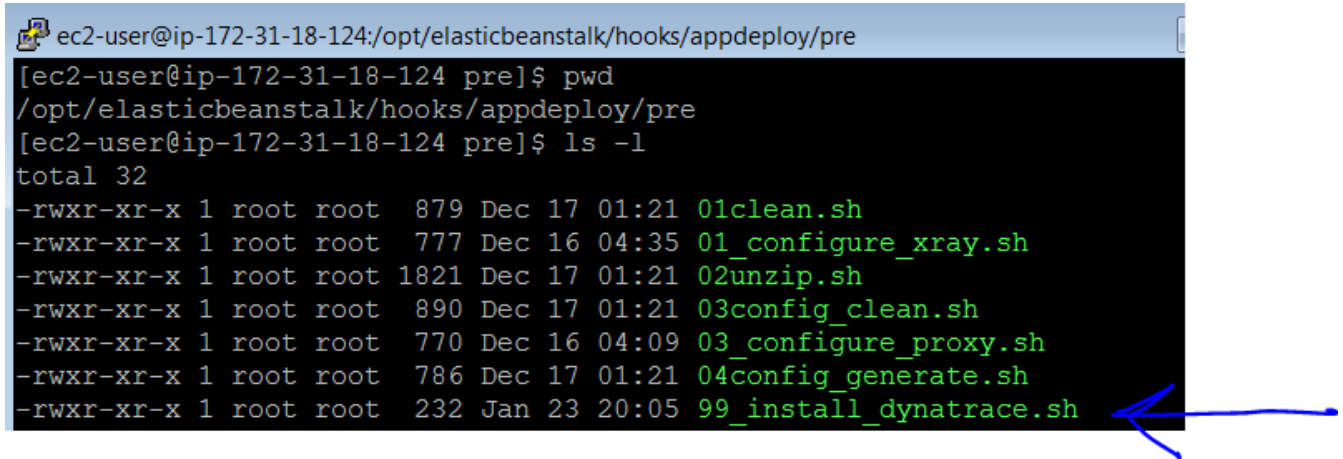


VALIDATE EBS APP AND AGENT INSTALL

3. Verify DT got installed into /opt folder

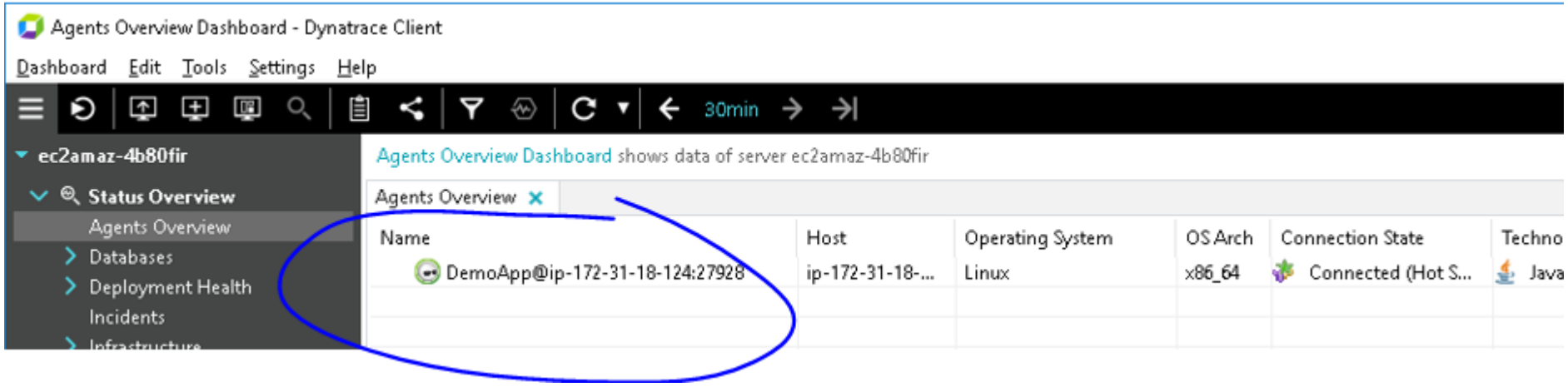


4. You can also see the shell scripts that EBS extension created in "/opt/elasticbeanstalk/hooks/appdeploy". Other files are from EBS. Custom scripts are in PRE folder. Scripts are run in alphabetical order.



VALIDATE EBS APP AND AGENT INSTALL

5. In Dynatrace, you should now see the agent

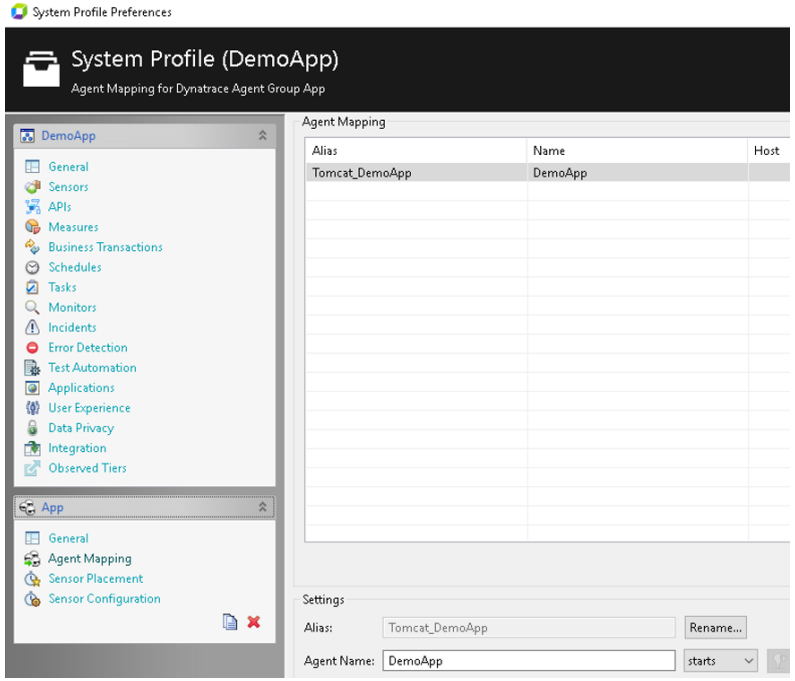


STEP 3

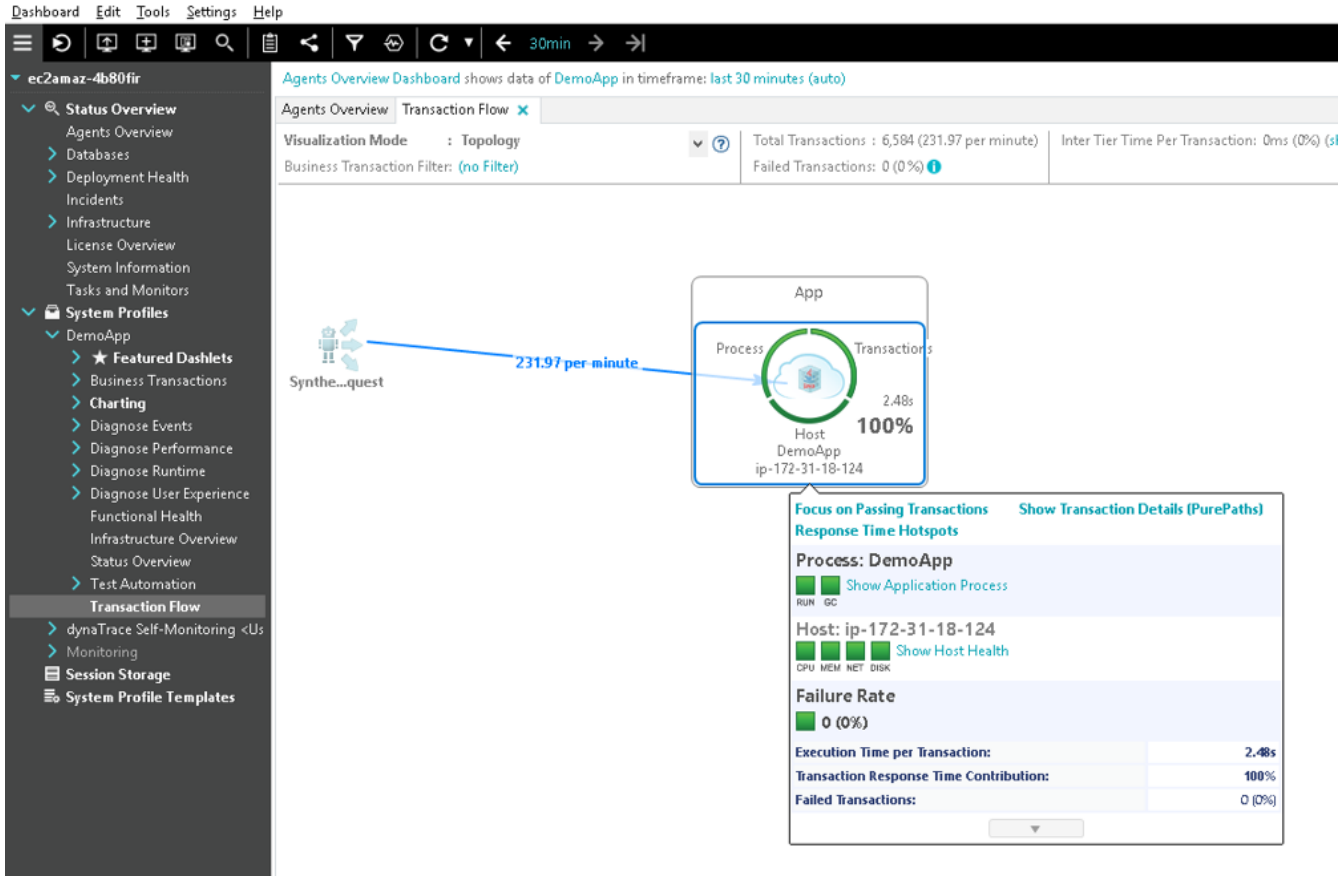
VERIFY SETUP

CONFIGURE DEMO APPLICATION PROFILE

1. Make a new System profile and add an App Tier using the prefix for the agent (see below)

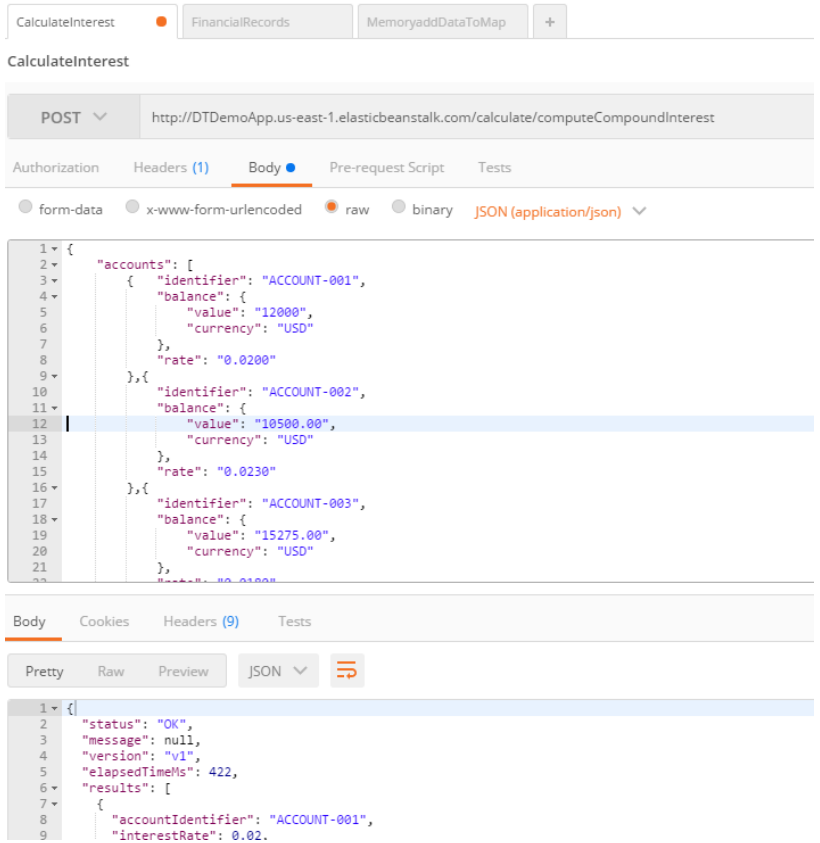


2. You can verify in monitoring dashboard (see right picture)



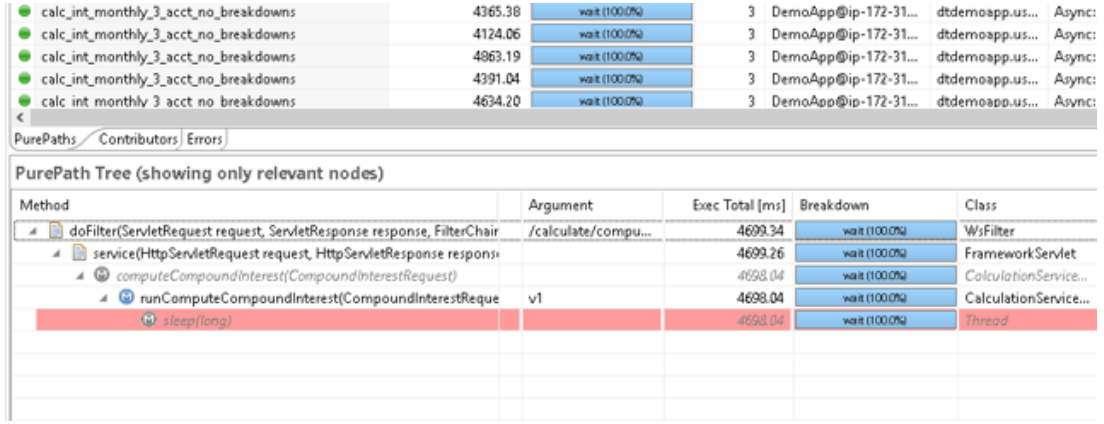
MAKE REQUESTS AND VERIFY PUREPATH

1. Quick test using “Chrome” postman App Add-in.



2. Or use the soapUI project within the source repo: https://github.com/jsicree/perf-demo/tree/master/demoweb/test

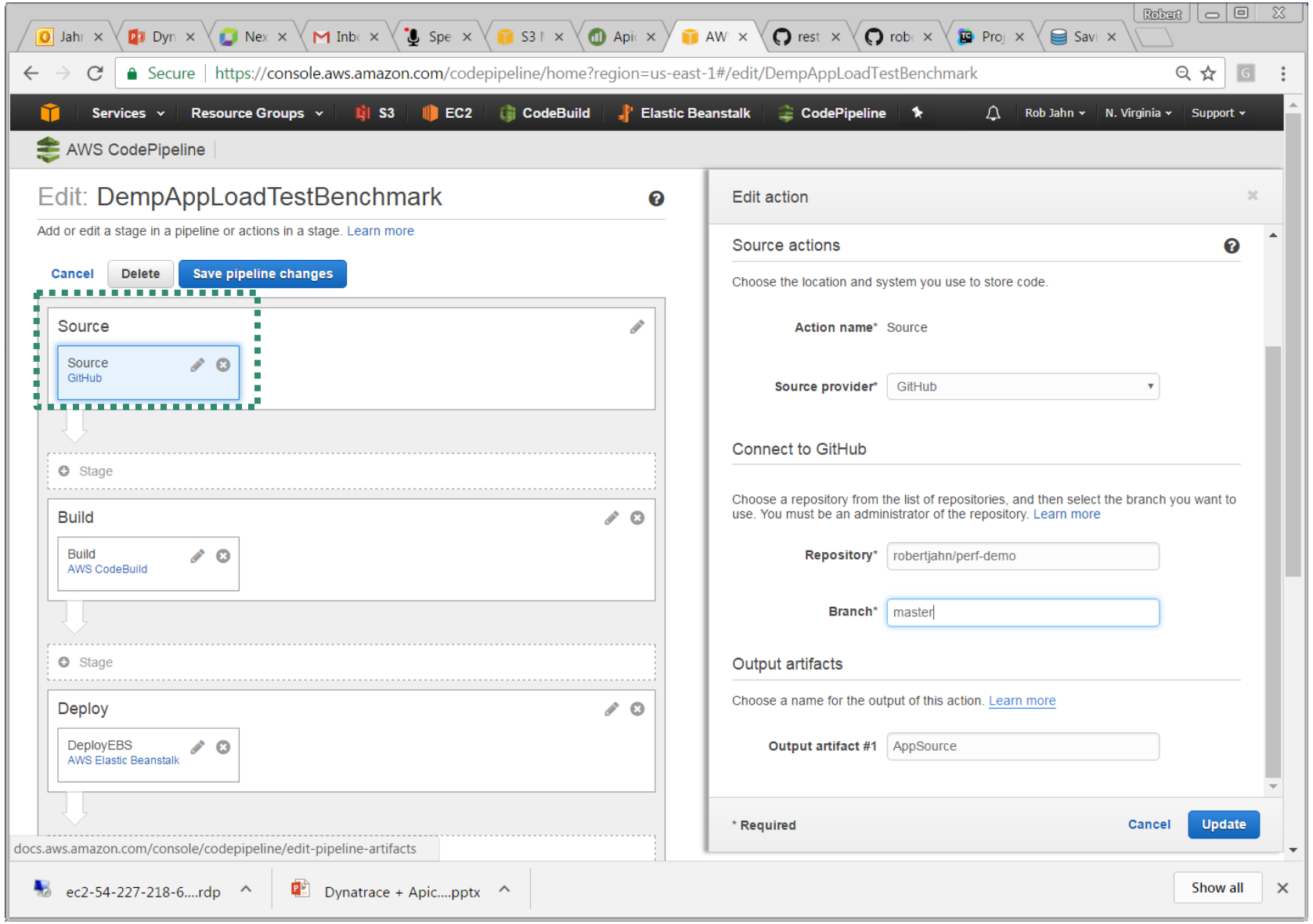
3. Purepath should now be visible



AWS CONFIGURATION DETAILS

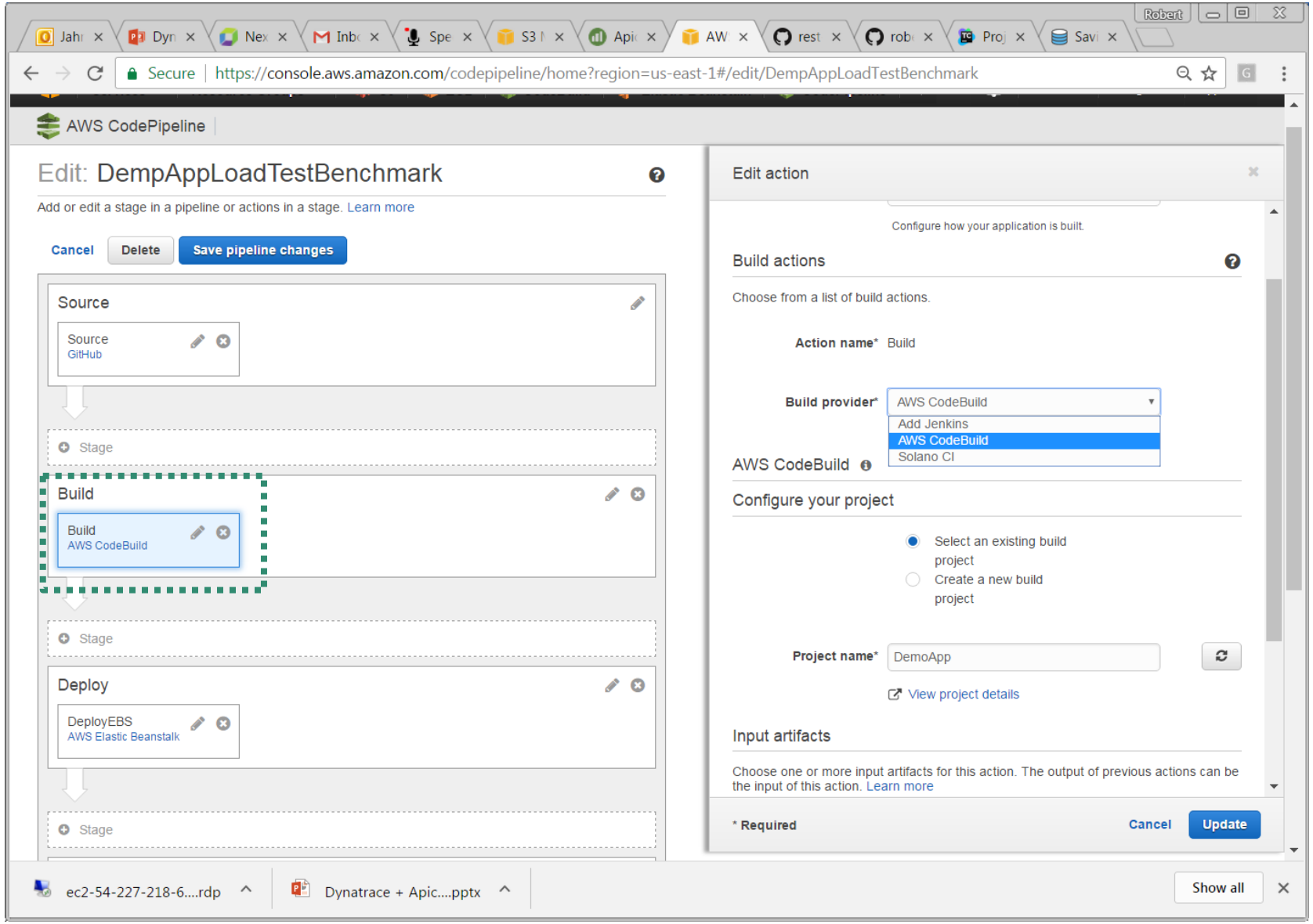
SOURCE STAGE IN AWS CODEPIPELINE

Commit Triggers the Pipeline!



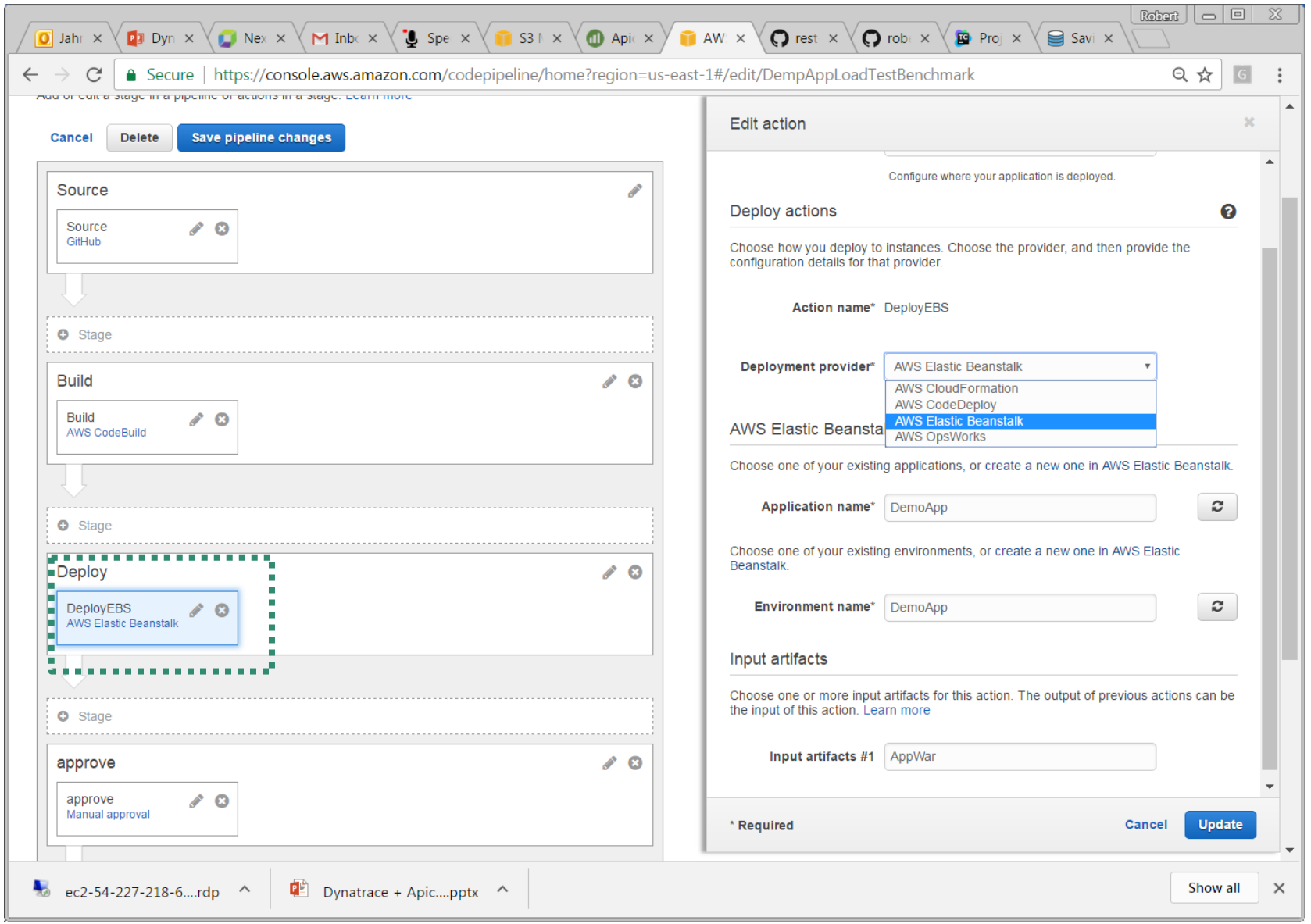
BUILD STAGE IN AWS CODEPIPELINE

Trigger AWS CodeBuild



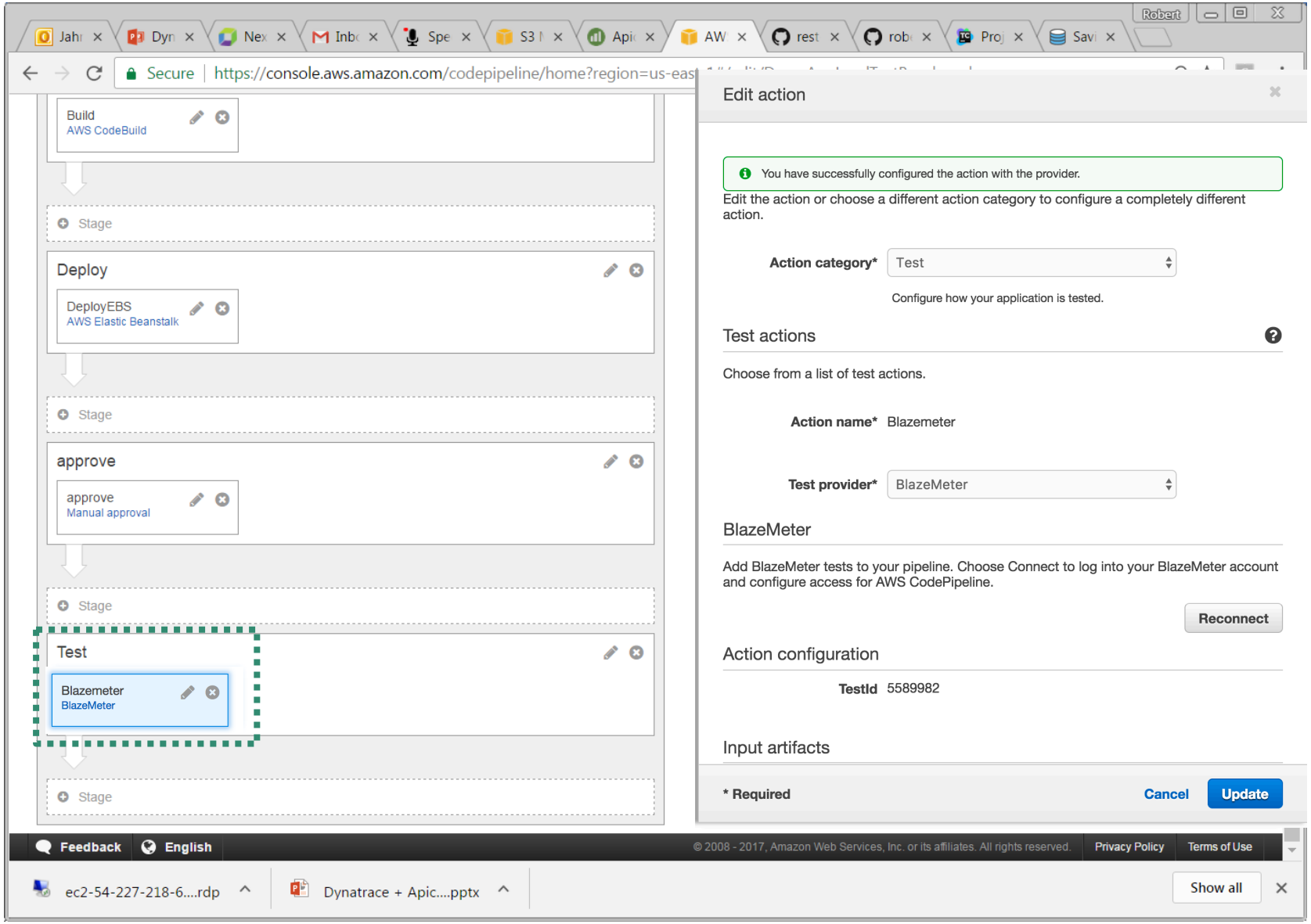
DEPLOY STAGE IN AWS CODEPIPELINE

Trigger AWS EBS Deployment



TEST STAGE IN AWS CODEPIPELINE

Trigger Blazemeter Test



TEST STAGE IN AWS CODEPIPELINE