

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

CMG Boston – April 2017



**Joe Sicree**

Sr. Consultant @ CGI

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



**Rob Jahn**

Technical Director @ Elyxor

Rob.Jahn@elyxor.com



# 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

**Edit inbound rules** [X]

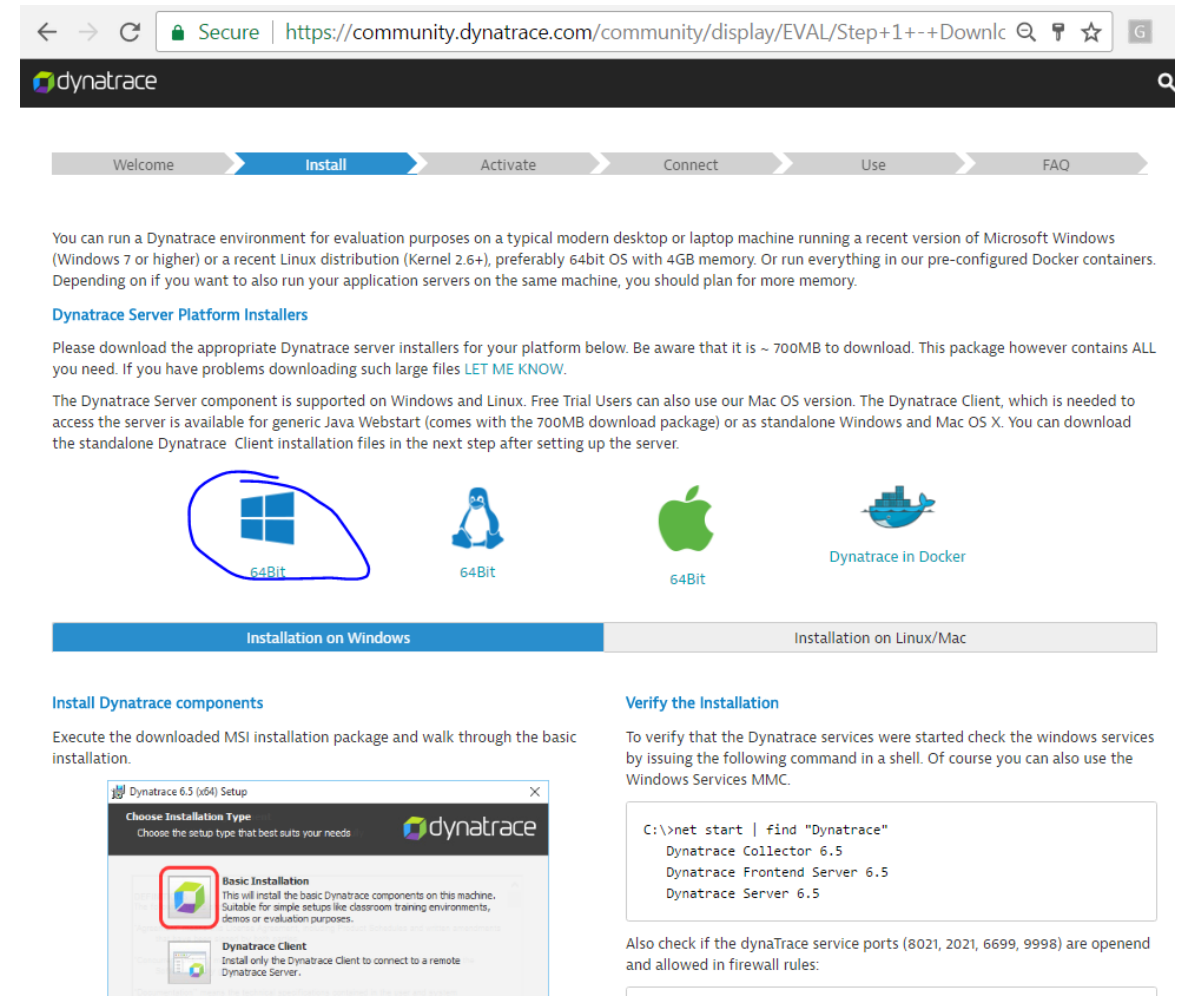
Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	
Custom TCP Rule ▼	TCP	9998	Custom ▼ 0.0.0.0/0	✕
RDP ▼	TCP	3389	Custom ▼ 24.91.36.140/32	✕

Add Rule Cancel Save

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



The screenshot shows a web browser at the URL <https://community.dynatrace.com/community/display/EVAL/Step+1+-+Downlc>. The page has a navigation bar with links: Welcome, Install (active), Activate, Connect, Use, and FAQ. Below the navigation bar, there is a text block explaining that Dynatrace can be run on Windows or Linux. It then lists "Dynatrace Server Platform Installers" and provides instructions on how to download the appropriate installer for the user's platform. There are four icons representing different installation methods: Windows 64Bit (circled in blue), Linux 64Bit, Mac 64Bit, and Dynatrace in Docker. Below the icons, there are two tabs: "Installation on Windows" (active) and "Installation on Linux/Mac". Under the "Installation on Windows" tab, there is a section titled "Install Dynatrace components" which instructs the user to execute the downloaded MSI installation package. Below this, there is a screenshot of the "Dynatrace 6.5 (x64) Setup" window, which shows the "Basic Installation" option selected. To the right of the "Installation on Windows" tab, there is a section titled "Verify the Installation" which provides instructions on how to verify the installation by checking the Windows services. It includes a code block with the command `C:\>net start | find "Dynatrace"` and the expected output: `Dynatrace Collector 6.5`, `Dynatrace Frontend Server 6.5`, and `Dynatrace Server 6.5`. Below this, there is a note about checking if the Dynatrace service ports (8021, 2021, 6699, 9998) are open and allowed in the firewall.

Secure | <https://community.dynatrace.com/community/display/EVAL/Step+1+-+Downlc>

Welcome Install Activate Connect Use FAQ

You can run a Dynatrace environment for evaluation purposes on a typical modern desktop or laptop machine running a recent version of Microsoft Windows (Windows 7 or higher) or a recent Linux distribution (Kernel 2.6+), preferably 64bit OS with 4GB memory. Or run everything in our pre-configured Docker containers. Depending on if you want to also run your application servers on the same machine, you should plan for more memory.

**Dynatrace Server Platform Installers**

Please download the appropriate Dynatrace server installers for your platform below. Be aware that it is ~ 700MB to download. This package however contains ALL you need. If you have problems downloading such large files [LET ME KNOW](#).

The Dynatrace Server component is supported on Windows and Linux. Free Trial Users can also use our Mac OS version. The Dynatrace Client, which is needed to access the server is available for generic Java Webstart (comes with the 700MB download package) or as standalone Windows and Mac OS X. You can download the standalone Dynatrace Client installation files in the next step after setting up the server.

64Bit 64Bit 64Bit Dynatrace in Docker

Installation on Windows Installation on Linux/Mac

**Install Dynatrace components**

Execute the downloaded MSI installation package and walk through the basic installation.

**Verify the Installation**

To verify that the Dynatrace services were started check the windows services by issuing the following command in a shell. Of course you can also use the Windows Services MMC.

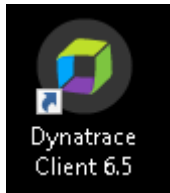
```
C:\>net start | find "Dynatrace"
Dynatrace Collector 6.5
Dynatrace Frontend Server 6.5
Dynatrace Server 6.5
```

Also check if the dynaTrace service ports (8021, 2021, 6699, 9998) are openend and allowed in firewall rules:

# 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

Name	Description	Status	Startup Type	L
Device Association Service	Enables pairing between the system and wired or wireless devices.		Manual (Trig...	L
Device Install Service	Enables a computer to recognize and adapt to hardware.		Manual (Trig...	L
Device Management Enrollment Service	Performs Device Enrollment Activities for Device Management.		Manual	L
Device Setup Manager	Enables the detection, download and installation of device drivers.		Manual (Trig...	L
DevQuery Background Discovery Broker	Enables apps to discover devices with a background discovery process.		Manual (Trig...	L
DHCP Client	Registers and updates IP addresses and DNS records.	Running	Automatic	L
Diagnostic Policy Service	The Diagnostic Policy Service enables problem detection and diagnosis.	Running	Automatic (D...	L
Diagnostic Service Host	The Diagnostic Service Host is used by the Diagnostic Policy Service.		Manual	L
Diagnostic System Host	The Diagnostic System Host is used by the Diagnostic Policy Service.	Running	Manual	L
Distributed Link Tracking Client	Maintains links between NTFS files within a computer.	Running	Automatic	L
Distributed Transaction Coordinator	Coordinates transactions that span multiple resource managers.	Running	Automatic (D...	L
dmwappushsvc	WAP Push Message Routing Service		Manual (Trig...	L
DNS Client	The DNS Client service (dnscache) caches Domain Name System (DNS) information.	Running	Automatic (T...	L
Downloaded Maps Manager	Windows service for application access to downloaded maps.		Automatic (D...	L
Dynatrace Collector 6.5	Dynatrace Collector 6.5	Running	Automatic	L
Dynatrace Frontend Server 6.5	Dynatrace Server 6.5	Running	Automatic	L
Dynatrace Memory Analysis Server 6.5	Dynatrace Memory Analysis Server 6.5		Manual	L
Dynatrace Server 6.5	Dynatrace Server 6.5	Running	Automatic	L
Dynatrace Web Server Agent 6.5	Dynatrace Web Server Agent 6.5		Manual	L
Embedded Mode	The Embedded Mode service enables scenarios related to embedded mode.		Manual (Trig...	L
Encrypting File System (EFS)	Provides the core file encryption technology used to encrypt files.		Manual (Trig...	L
Enterprise App Management Service	Enables enterprise application management.		Manual	L

ec2amaz-4b80fir

Connectivity

License

Settings

Services

Collectors

Performance Warehouse

Sensor Packs

Users

Plugins

Updates

Storage

Infrastructure

Databases

Geographical Locations

MQ Tagging

Synthetic Monitoring

Realtime Streaming

Certificate Management

Information

Assignments

UEM Volume Reservation

General

✓

The current license is set up properly. Refer to the [licensing documentation](#) and [license management website](#) for more information.

License:

201701171009 - Test Center Edition

Licensed to:

Free Trial

Valid:

from Jan 15, 2017 to Feb 17, 2017  
(will continue to work with local agent connections)

External Id:

0017000000:TAHcAAO

Technology	Licensed Units	Unit
Java	5	Agents
.NET WOSIs	5	OS instances
Native	1	Agents
Browser	2	Agents
Web Server	5	Agents/IIS instances
PHP	5	Agents
Host Monitoring	1	Agents
Node.js	5	Agents
Database	1	Agents (8 CPU Cores)

UEM and Consumption

✓

Remaining UEM volume:

99,999

Next volume renewal:

-

Import License

Delete License

Activate License

EULA

Open in Dashboard

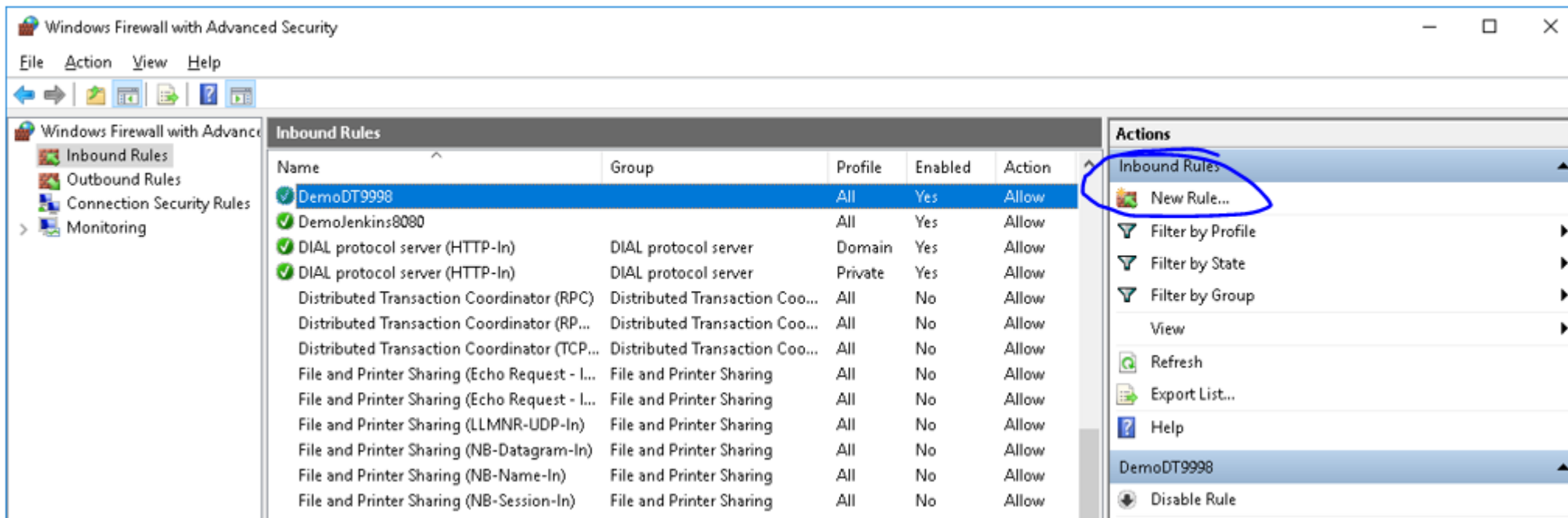
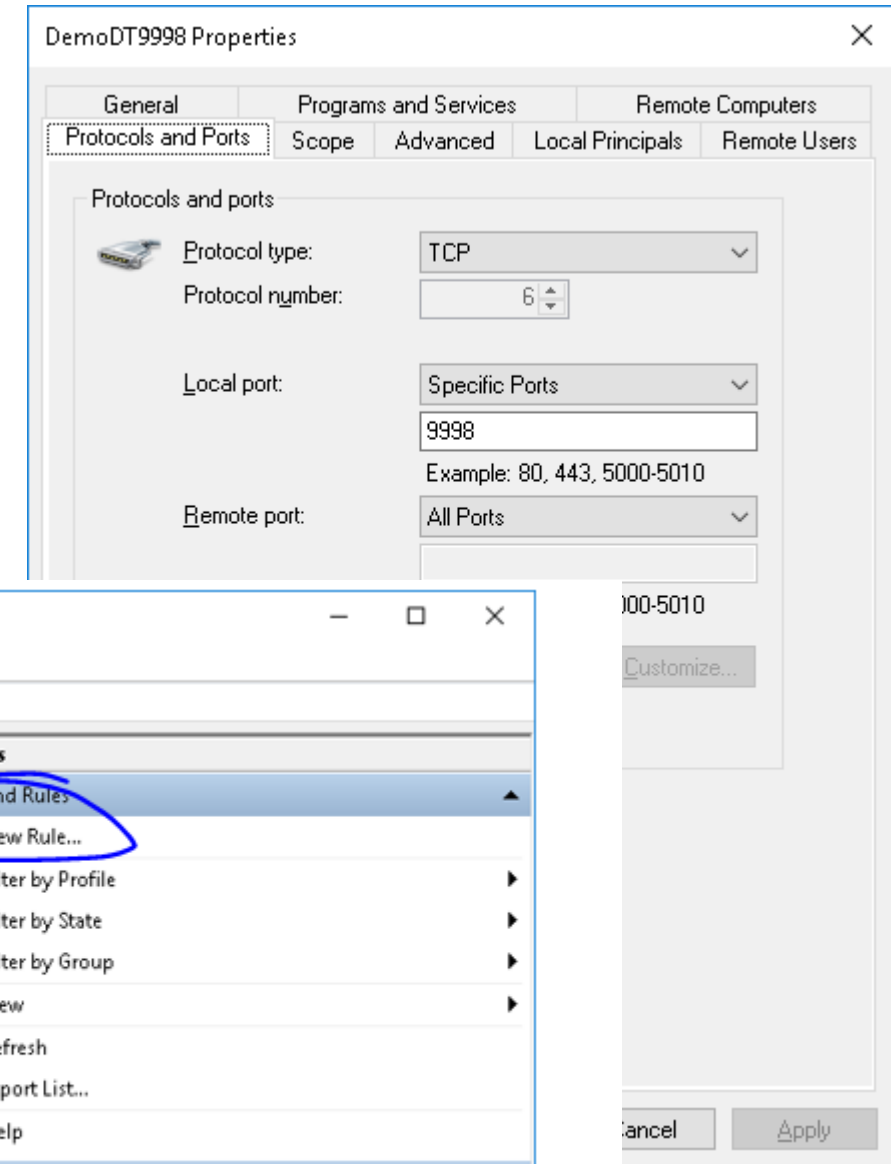
Add Volume

Deactivate Volume

UEM Free Trial

# 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

The screenshot shows the AWS CodeBuild console interface. The browser address bar displays the URL: <https://console.aws.amazon.com/codebuild/home?region=us-east-1#/projects/DemoApp/updates>. The page has a sidebar with 'Build projects' and 'Build history' tabs. The main content area is titled 'Project' and contains the following configuration sections:

- Project name\***: DemoApp
- Description**: + Add description
- Source: What to build**
  - Current source**: <https://github.com/robertjahn/perf-demo.git> (with an 'Update source' link)
- Environment: How to build**
  - Current image**: aws/codebuild/java:openjdk-8 (with an 'Update image' link)
  - Current build specification**: Using buildspec.yml in the source code root directory (with an 'Update build specification' link)
- Artifacts: Where to put the artifacts from this build project**
  - Artifacts type\***: Amazon S3 (with an info icon)
  - Artifacts name**: DemoApp (with an info icon)
  - Bucket name\***: rjahn-temp-s3 (with a dropdown arrow)
- Service role**
  - Specify a service role that enables AWS CodeBuild to call dependent AWS services on your behalf. [Learn more.](#)
  - ☐ Create a role
  - ☒ Choose an existing service role from your account
  - Role name\***: codebuild-DemoApp-service-role (with a dropdown arrow)

# CODE BUILD

## 1. Successful build will look like this

DemoApp:c3ff3515-854d-4ff3-ac6f-8a1a60bcdf8 **Succeeded**

Review your build details as it progresses.

Build

Build ARN	arn:aws:codebuild:us-east-1:929478327752:build/DemoApp:c3ff3515-854d-4ff3-ac6f-8a1a60bcdf8
Build project	DemoApp
Source provider	Git-Hub
Repository	<a href="https://github.com/robertjahn/perf-demo.git">https://github.com/robertjahn/perf-demo.git</a>
Start time	6 minutes ago
End time	2 minutes ago
Status	<b>Succeeded</b>
Initiator	root

► Build details

Phase details

Name	Status
► SUBMITTED	Succeeded
► PROVISIONING	Succeeded
► DOWNLOAD_SOURCE	Succeeded
► INSTALL	Succeeded
► PRE_BUILD	Succeeded
► BUILD	Succeeded
► POST_BUILD	Succeeded
► UPLOAD_ARTIFACTS	Succeeded
► FINALIZING	Succeeded
► COMPLETED	Succeeded

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

Services ▾ Resource Groups ▾ S3 EC2

Upload Create Folder Actions ▾ Versions: Hide Show

All Buckets / rjahn-temp-s3

	Name
<input type="checkbox"/>	AWSCodePipeline-S3-AWSCodeDeploy_Windows.zip
<input checked="" type="checkbox"/>	DemoApp
<input type="checkbox"/>	DemoApp

## 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

## Create a new environment

Launch an environment with a sample application or your own code. By creating an environment, you allow permissions on your behalf. [Learn more](#)

**Application name** DemoApp

**Tier** Web Server ([Choose tier](#))

**Platform** Tomcat ▼

Choose [Configure more options](#) for more platform configuration options.

**Application code** ☐ Sample application

Get started right away with sample code.

☐ Existing version

Application versions that you have uploaded for **DemoApp**.

-- Choose a version -- ▼

☒ Upload your code

Upload a source bundle from your computer or copy one from Amazon S3.

 **Upload**

ZIP or WAR

# EBS DOMAIN

## Configure LowCost-env

Start from a preset that matches your use case or choose *Custom configuration*

**Configuration presets**

- ☒ Low cost (*Free Tier eligible*)
- ☐ High availability
- ☐ Custom configuration

Platform 64bit Amazon Linux 2016.09 v2.5.1 running Tomcat 8 Java 8 [Change](#)

### Environment settings

Name: LowCost-env  
Domain: autogenerated  
Description: blank  
Tags: none

[Modify](#)

### Capacity

Environment type: single instance  
Availability Zones: Any  
Instances: 1-1

[Modify](#)

### Security

Service role: aws-elasticbeanstalk-service-role  
Virtual machine key pair: --  
Virtual machine instance profile: aws-elasticbeanstalk-ec2-role

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.

## Environment settings

Choose the name, subdomain, and description for your environment. Environment set

### Parameters

**Name** Dev

**Domain** MyDemo1

MyDemo1.us-east-1.elasticbeanstalk.com **is available**

[Check availability](#)

## Security

### Service role

**Service role** aws-elasticbeanstalk-service-role

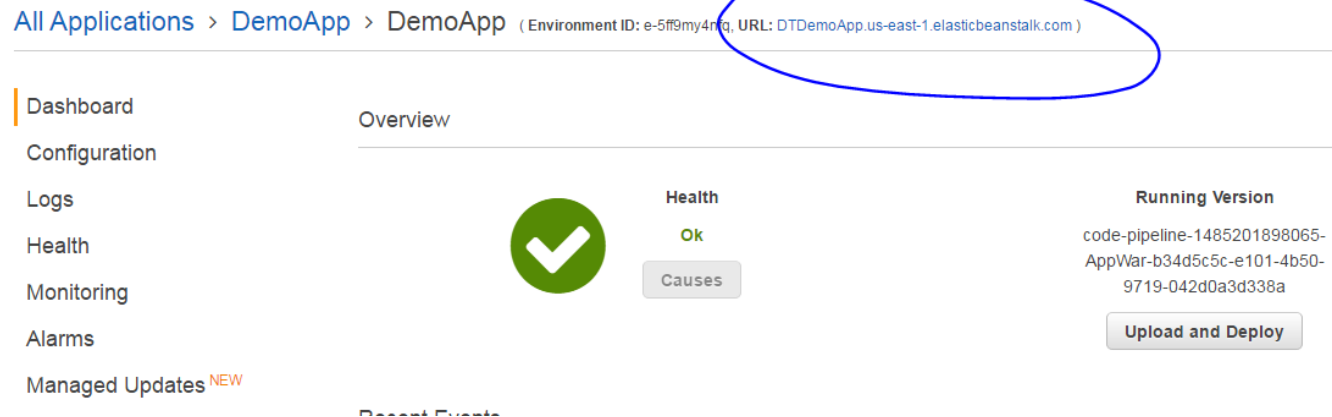
### Virtual machine permissions

**EC2 key pair** PipelineDemo

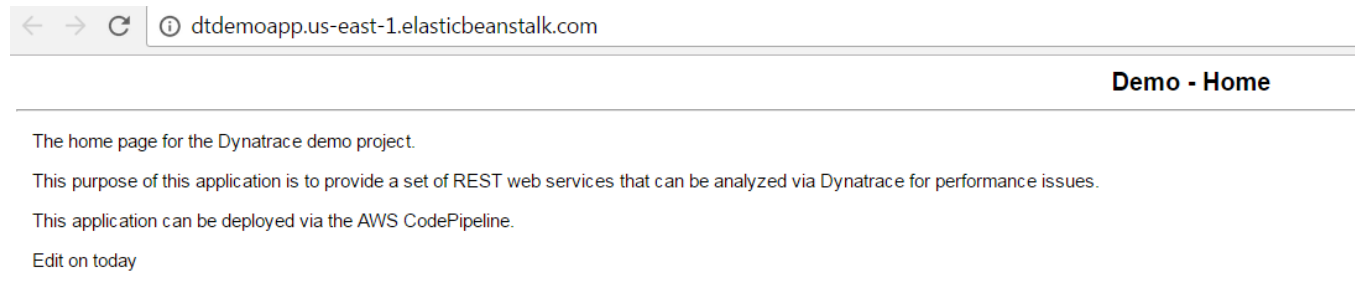
**IAM instance profile** aws-elasticbeanstalk-ec2-role

# EBS DOMAIN

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

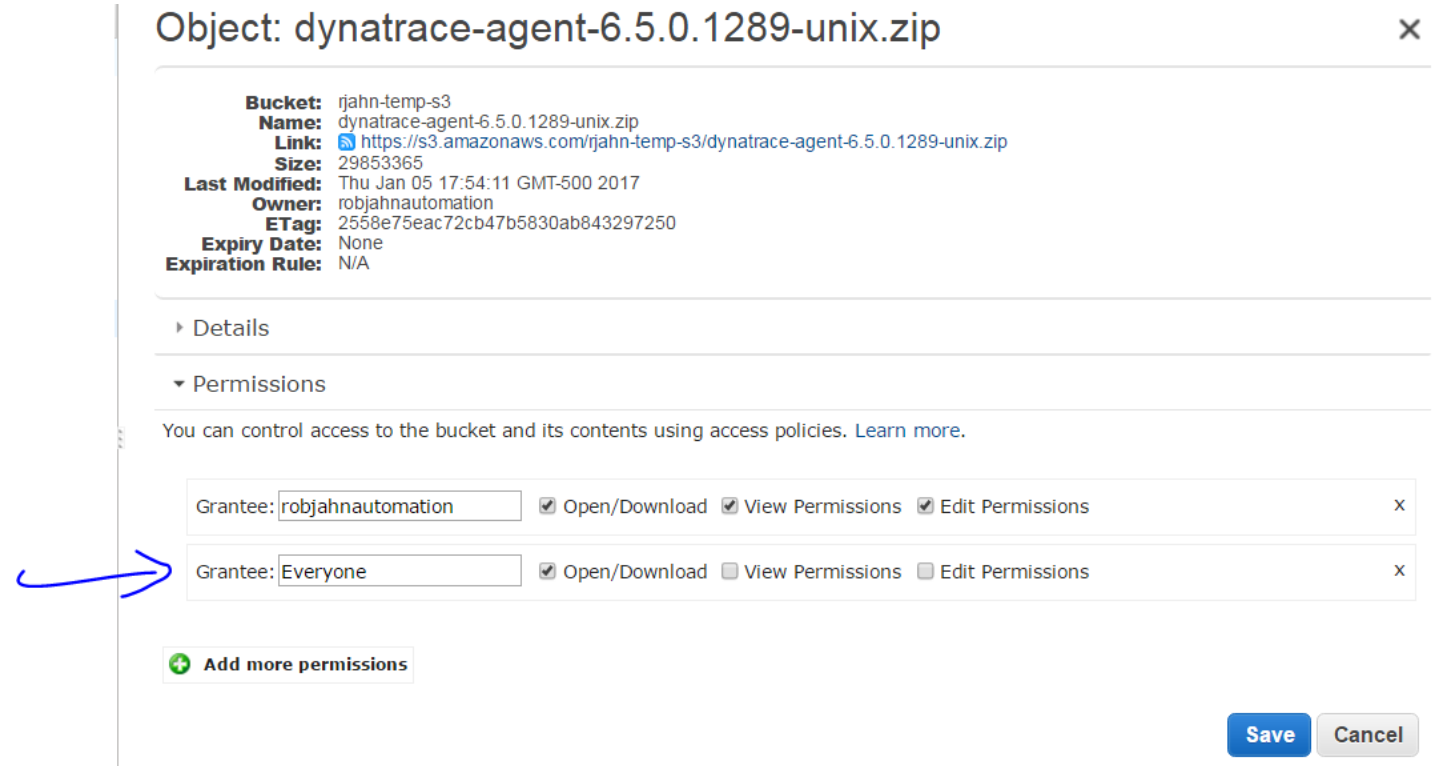


2. App should show App home page in Browser



# 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



Object: dynatrace-agent-6.5.0.1289-unix.zip

**Bucket:** rjahn-temp-s3  
**Name:** dynatrace-agent-6.5.0.1289-unix.zip  
**Link:** <https://s3.amazonaws.com/rjahn-temp-s3/dynatrace-agent-6.5.0.1289-unix.zip>  
**Size:** 29853365  
**Last Modified:** Thu Jan 05 17:54:11 GMT-500 2017  
**Owner:** robjahnautomation  
**ETag:** 2558e75eac72cb47b5830ab843297250  
**Expiry Date:** None  
**Expiration Rule:** N/A

Details

Permissions

You can control access to the bucket and its contents using access policies. [Learn more.](#)

Grantee: robjahnautomation	<input checked="" type="checkbox"/> Open/Download	<input checked="" type="checkbox"/> View Permissions	<input checked="" type="checkbox"/> Edit Permissions	x
Grantee: Everyone	<input checked="" type="checkbox"/> Open/Download	<input type="checkbox"/> View Permissions	<input type="checkbox"/> Edit Permissions	x

+ Add more permissions

Save Cancel

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

28 lines (25 sloc) | 863 Bytes





```
1  files:
2    "/tmp/dynatrace-agent-6.5.0.1289-unix.zip":
3      mode: "000755"
4      owner: root
5      group: root
6      source: https://s3.amazonaws.com/rjahn-temp-s3/dynatrace-agent-6.5.0.1289-unix.zip
7
8    "/opt/elasticbeanstalk/hooks/appdeploy/pre/99_install_dynatrace.sh":
9      mode: "000755"
10     owner: root
11     group: root
12     content: |
13         #!/bin/bash
14         if [ ! -d /opt/dynatrace-6.5 ]
15         then
16             #unzip_dynatrace
17             sudo unzip -o /tmp/dynatrace-agent-6.5.0.1289-unix.zip -d /tmp
18             #install_dynatrace in silent mode
19             cd /opt
20             sudo java -jar /tmp/dynatrace-agent-6.5.0.1289-unix.jar -y
21             fi
22
23  commands:
24    install_dynatrace:
25      command: /opt/elasticbeanstalk/hooks/appdeploy/pre/99_install_dynatrace.sh
26      cwd: /opt
27      test: "[ -d /opt/dynatrace-6.5 ]"
```

Branch: master ▾ perf-demo / demoweb / .ebextensions /

This branch is 96 commits ahead, 8 commits behind jsicree:master.

 robertjahn committed on GitHub Update 04-tomcat.config

..

 01_install_DT.config	Update 01_install_DT.config
 02-move-war.config	Rename move-war.config to 02-move-war
 03-run-expand-war.config	Rename 03-run-expand-war to 03-run-e
 04-tomcat.config	Update 04-tomcat.config



# EBS CONFIGURATION

## DT agent coming in from EBS

Agents Overview Dashboard - Dynatrace Client

Dashboard Edit Tools Settings Help

ec2amaz-4b80fir

▼ Status Overview

Agents Overview

> Databases

> Deployment Health

Incidents

> Infrastructure

License Overview

System Information

Tasks and Monitors

▼ System Profiles

> DemoApp

> DemoApp

> DemoApp

> DemoApp

> DemoApp

> DemoApp

> DemoApp

> DemoApp

> DemoApp

> DemoApp

> DemoApp

> DemoApp

> DemoApp

> DemoApp

> DemoApp

> DemoApp

Agents Overview Dashboard shows data of server ec2amaz-4b80fir

Agents Overview x

Name	Host	Operating System	OS Arch	Connection State	Technology
DemoApp@ip-172-31-18-124:27928	ip-172-31-18-...	Linux	x86_64	Connected (Hot S...	Java

Branch: master

perf-demo / demoweb / .ebextensions / 04-tomcat.config

Find file

Copy path

robertjahn Update 04-tomcat.config

b4bab34 44 minutes ago

1 contributor

8 lines (6 sloc) | 416 Bytes

Raw

Blame

History

```
1 # the DT servername will need to be updated for you server
2 # if you use EC2 with no fixed domain then you also need to edit it after each stop/start of the instance
3
4 option_settings:
5   - namespace: aws:elasticbeanstalk:container:tomcat:jvmoptions
6     option_name: JVM Options
7     value: -agentpath:/opt/dynatrace-6.5/agent/lib64/libdtagent.so=name=DemoApp,collector=ec2-54-227-218-63.compute-1.amazonaws.com:9998
```

# 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

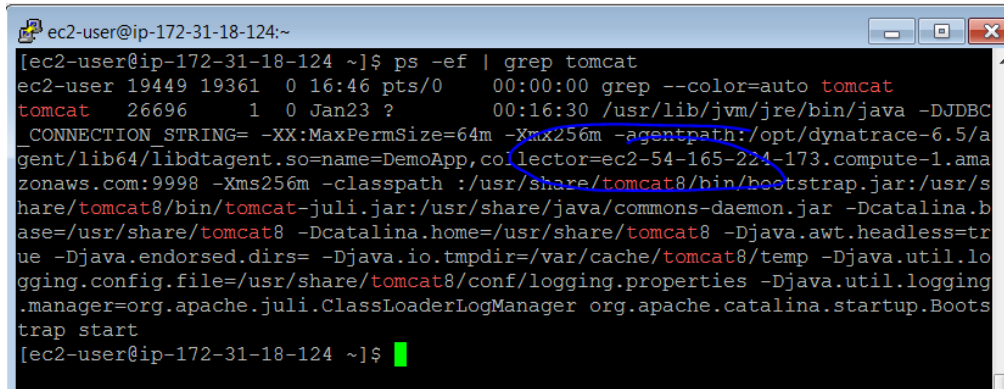
The image shows two screenshots of GitHub configuration files. The first screenshot is for the file `01_install_DT.config` in the `perf-demo / demoweb / .ebextensions` directory. It shows a commit by Joseph Sicree. The file content includes a `files:` section with a `source` value: `https://s3.amazonaws.com/rjahn-temp-s3/dynatrace-agent-6.5.0.1289-unix.zip`, which is circled in blue. The second screenshot is for the file `04-tomcat.config` in the same directory. It shows a commit by jsicree. The file content includes an `option_settings:` section with a `value` that contains a collector URL: `collector=ec2-54-89-79-162.compute-1.amazonaws.com:9998`, which is also circled in blue.

```
Branch: master perf-demo / demoweb / .ebextensions / 01_install_DT.config
Joseph Sicree Added eb extensions
0 contributors
28 lines (25 sloc) 863 Bytes
1 files:
2 "/tmp/dynatrace-agent-6.5.0.1289-unix.zip":
3   mode: "000755"
4   owner: root
5   group: root
6   source: https://s3.amazonaws.com/rjahn-temp-s3/dynatrace-agent-6.5.0.1289-unix.zip
7
8   "/opt/elasticbeanstalk/hooks/addon/ops/01_install_dynatrace.sh":

Branch: master perf-demo / demoweb / .ebextensions / 04-tomcat.config
jsicree Updated DT ip 7051779 6 days ago
1 contributor
8 lines (6 sloc) 415 Bytes
1 # the DT servername will need to be updated for you server
2 # if you use EC2 with no fixed domain then you also need to edit it after each stop/start of the instance
3
4 option_settings:
5   - namespace: aws:elasticbeanstalk:container:tomcat:jvmoptions
6     option_name: JVM Options
7     value: -agentpath:/opt/dynatrace-6.5/agent/lib64/libdtagent.so=name=DemoApp,collector=ec2-54-89-79-162.compute-1.amazonaws.com:9998
```

# 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`




```
ec2-user@ip-172-31-18-124:~  
[ec2-user@ip-172-31-18-124 ~]$ ps -ef | grep tomcat  
ec2-user 19449 19361  0 16:46 pts/0    00:00:00 grep --color=auto tomcat  
tomcat   26696      1  0 Jan23 ?        00:16:30 /usr/lib/jvm/jre/bin/java -DJDBC  
_CONNECTION_STRING= -XX:MaxPermSize=64m -Xmx256m -agentpath:/opt/dynatrace-6.5/a  
gent/lib64/libdtagent.so=name=DemoApp,collector=ec2-54-165-224-173.compute-1.ama  
zonaws.com:9998 -Xms256m -classpath :/usr/share/tomcat8/bin/bootstrap.jar:/usr/s  
hare/tomcat8/bin/tomcat-juli.jar:/usr/share/java/commons-daemon.jar -Dcatalina.b  
ase=/usr/share/tomcat8 -Dcatalina.home=/usr/share/tomcat8 -Djava.awt.headless=tr  
ue -Djava.endorsed.dirs= -Djava.io.tmpdir=/var/cache/tomcat8/temp -Djava.util.lo  
gging.config.file=/usr/share/tomcat8/conf/logging.properties -Djava.util.logging  
.manager=org.apache.juli.ClassLoaderLogManager org.apache.catalina.startup.Boots  
trap start  
[ec2-user@ip-172-31-18-124 ~]$
```

# VALIDATE EBS APP AND AGENT INSTALL


## 3. Verify DT got installed into /opt folder

```
ec2-user@ip-172-31-18-124:~  
[ec2-user@ip-172-31-18-124 ~]$ ls -l /opt  
total 12  
drwxr-xr-x  5 root root 4096 Sep 23 10:01 aws  
drwxr-xr-x  5 root root 4096 Jan 19 03:16 dynatrace-6.5  
drwxr-xr-x 11 root root 4096 Jan 19 03:16 elasticbeanstalk  
[ec2-user@ip-172-31-18-124 ~]$
```



## 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.




```
ec2-user@ip-172-31-18-124:/opt/elasticbeanstalk/hooks/appdeploy/pre  
[ec2-user@ip-172-31-18-124 pre]$ pwd  
/opt/elasticbeanstalk/hooks/appdeploy/pre  
[ec2-user@ip-172-31-18-124 pre]$ ls -l  
total 32  
-rwxr-xr-x 1 root root  879 Dec 17 01:21 01clean.sh  
-rwxr-xr-x 1 root root  777 Dec 16 04:35 01_configure_xray.sh  
-rwxr-xr-x 1 root root 1821 Dec 17 01:21 02unzip.sh  
-rwxr-xr-x 1 root root  890 Dec 17 01:21 03config_clean.sh  
-rwxr-xr-x 1 root root  770 Dec 16 04:09 03_configure_proxy.sh  
-rwxr-xr-x 1 root root  786 Dec 17 01:21 04config_generate.sh  
-rwxr-xr-x 1 root root  232 Jan 23 20:05 99_install_dynatrace.sh
```



# VALIDATE EBS APP AND AGENT INSTALL

5. In Dynatrace, you should now see the agent

The screenshot displays the 'Agents Overview Dashboard' in the Dynatrace Client. The left sidebar shows a navigation menu with 'Status Overview' expanded, highlighting 'Agents Overview'. The main panel shows a table of agents. A blue circle highlights the first agent, 'DemoApp@ip-172-31-18-124:27928', which is connected and running on Linux with Java technology.

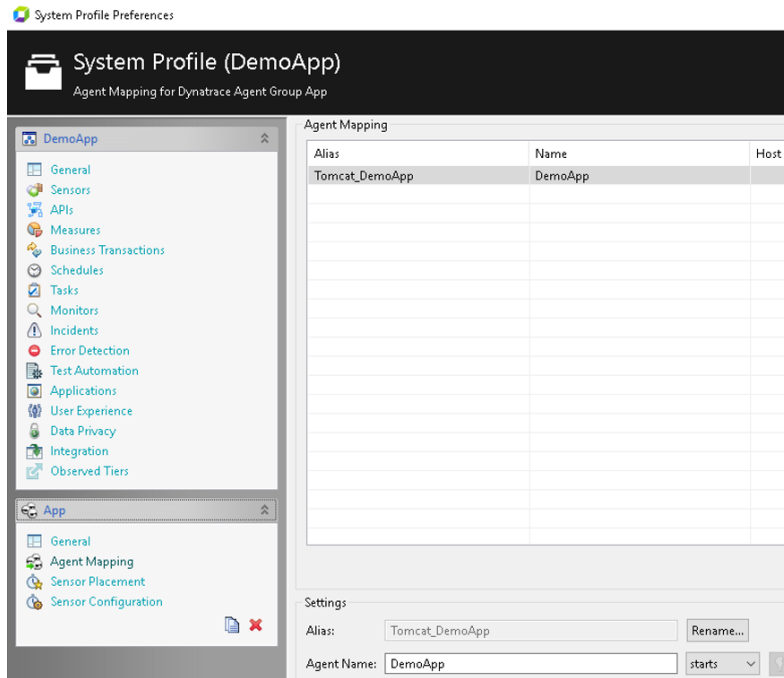
Name	Host	Operating System	OS Arch	Connection State	Techno
 DemoApp@ip-172-31-18-124:27928	ip-172-31-18-...	Linux	x86_64	 Connected (Hot S...	 Java

## STEP 3

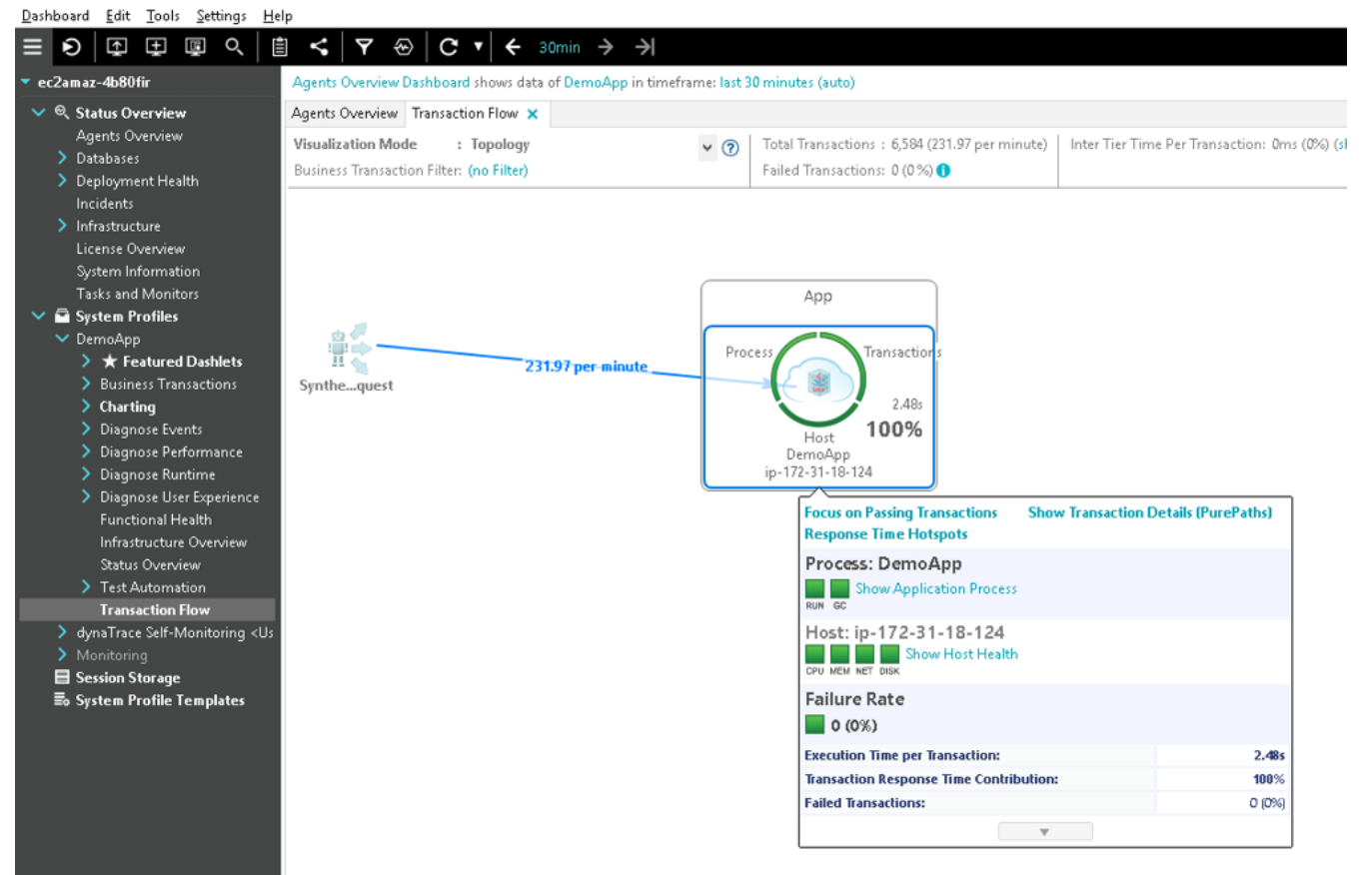
VERIFY SETUP

# CONFIGURE DEMO APPLICATION PROFILE

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

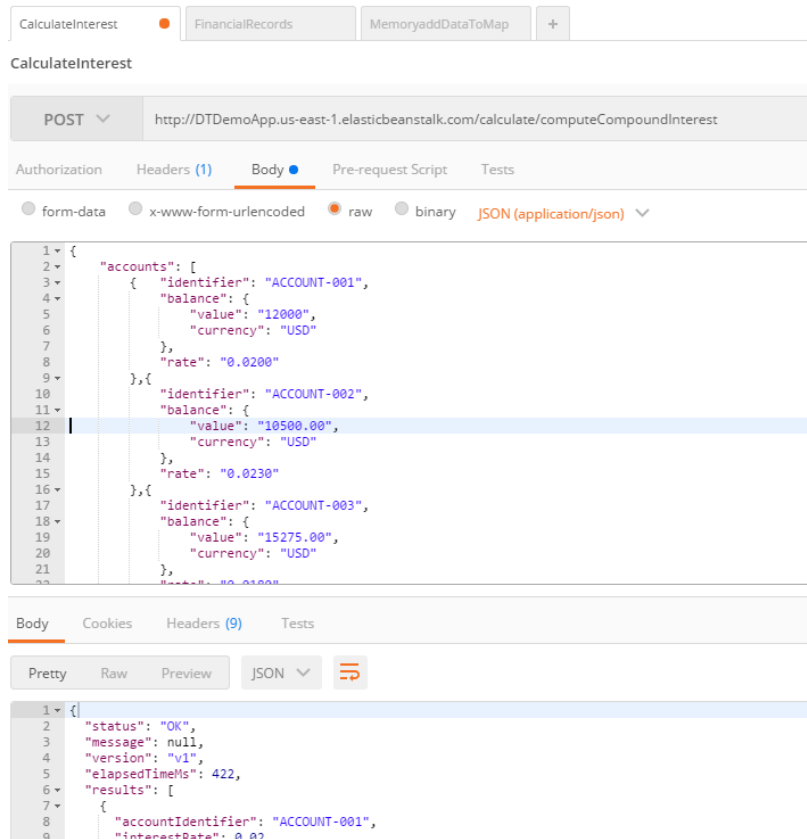


2. You can verify in monitoring dashboard (see right picture)



# 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

calc_int_monthly_3_acct_no_breakdowns	4365.38	wait (100.0%)	3	DemoApp@ip-172-31...	dttdemoapp.us...	Async:
calc_int_monthly_3_acct_no_breakdowns	4124.06	wait (100.0%)	3	DemoApp@ip-172-31...	dttdemoapp.us...	Async:
calc_int_monthly_3_acct_no_breakdowns	4863.19	wait (100.0%)	3	DemoApp@ip-172-31...	dttdemoapp.us...	Async:
calc_int_monthly_3_acct_no_breakdowns	4391.04	wait (100.0%)	3	DemoApp@ip-172-31...	dttdemoapp.us...	Async:
calc_int_monthly_3_acct_no_breakdowns	4634.20	wait (100.0%)	3	DemoApp@ip-172-31...	dttdemoapp.us...	Async:

PurePath Tree (showing only relevant nodes)					
Method	Argument	Exec Total [ms]	Breakdown	Class	
doFilter(ServletRequest request, ServletResponse response, FilterChain chain)	/calculate/compu...	4699.34	wait (100.0%)	WsFilter	
service(HttpServletRequest request, HttpServletResponse response)		4699.26	wait (100.0%)	FrameworkServlet	
computeCompoundInterest(CompoundInterestRequest request)		4698.04	wait (100.0%)	CalculationService...	
runComputeCompoundInterest(CompoundInterestRequest request)	v1	4698.04	wait (100.0%)	CalculationService...	
sleep(long)		4698.04	wait (100.0%)	Thread	





# AWS CONFIGURATION DETAILS

# ***SOURCE STAGE*** IN AWS CODEPIPELINE

Commit Triggers  
the Pipeline!

The screenshot displays the AWS CodePipeline console interface for editing a pipeline named 'DempAppLoadTestBenchmark'. The pipeline consists of three stages: 'Source', 'Build', and 'Deploy'. The 'Source' stage is highlighted with a green dashed border, indicating it is the current focus. The 'Build' stage uses the 'AWS CodeBuild' provider, and the 'Deploy' stage uses the 'AWS Elastic Beanstalk' provider. A right-hand panel titled 'Edit action' provides configuration details for the 'Source' stage action. It includes fields for 'Action name' (Source), 'Source provider' (GitHub), 'Repository' (robertjahn/perf-demo), 'Branch' (master), and 'Output artifact #1' (AppSource). The interface also shows buttons for 'Cancel', 'Delete', and 'Save pipeline changes' at the top, and a 'Show all' button at the bottom right.

# ***BUILD STAGE*** IN AWS CODEPIPELINE



Browser tabs: Jahi x, Dyn x, Nex x, Inbo x, Spe x, S3 x, Api x, AW x, rest x, rob x, Proj x, Savi x

URL: <https://console.aws.amazon.com/codepipeline/home?region=us-east-1#/edit/DempAppLoadTestBenchmark>

### AWS CodePipeline

## Edit: DempAppLoadTestBenchmark

Add or edit a stage in a pipeline or actions in a stage. [Learn more](#)

[Cancel](#) [Delete](#) [Save pipeline changes](#)

**Source**

Source  
GitHub

↓

**Build**

Build  
AWS CodeBuild

↓

**Deploy**

DeployEBS  
AWS Elastic Beanstalk

↓

### Edit action

Configure how your application is built.

**Build actions**

Choose from a list of build actions.

**Action name\*** Build

**Build provider\*** AWS CodeBuild

**AWS CodeBuild**

**Configure your project**

☒ Select an existing build project

☐ Create a new build project

**Project name\*** DemoApp [View project details](#)

**Input artifacts**

Choose one or more input artifacts for this action. The output of previous actions can be the input of this action. [Learn more](#)

**\* Required** [Cancel](#) [Update](#)

Taskbar: ec2-54-227-218-6....rdp, Dynatrace + Apic....pptx

# DEPLOY STAGE IN AWS CODEPIPELINE



Cancel Delete Save pipeline changes

Source

Source  
GitHub

Stage

Build

Build  
AWS CodeBuild

Stage

Deploy

DeployEBS  
AWS Elastic Beanstalk

Stage

approve

approve  
Manual approval

Edit action

Configure where your application is deployed.

Deploy actions

Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

Action name\* DeployEBS

Deployment provider\*

- AWS Elastic Beanstalk
- AWS CloudFormation
- AWS CodeDeploy
- AWS Elastic Beanstalk
- AWS OpsWorks

AWS Elastic Beanstalk

Choose one of your existing applications, or create a new one in AWS Elastic Beanstalk.

Application name\* DemoApp

Choose one of your existing environments, or create a new one in AWS Elastic Beanstalk.

Environment name\* DemoApp

Input artifacts

Choose one or more input artifacts for this action. The output of previous actions can be the input of this action. [Learn more](#)

Input artifacts #1 AppWar

\* Required Cancel Update

Show all

# TEST STAGE IN AWS CODEPIPELINE

Trigger Blazemeter Test

The screenshot displays the AWS CodePipeline console interface. On the left, a pipeline diagram shows four stages: 'Build' (using AWS CodeBuild), 'Deploy' (using DeployEBS on AWS Elastic Beanstalk), 'approve' (using Manual approval), and 'Test' (using Blazemeter). The 'Test' stage is highlighted with a green dashed border. On the right, the 'Edit action' modal is open for the 'Blazemeter' action. The modal shows the 'Action category' set to 'Test' and the 'Test provider' set to 'BlazeMeter'. Below this, the 'BlazeMeter' section provides instructions to add tests and connect to the account, with a 'Reconnect' button. The 'Action configuration' section shows the 'TestId' as '5589982'. At the bottom of the modal are 'Cancel' and 'Update' buttons. The footer of the console shows the user 'Robert', a feedback link, language settings, and copyright information.

**Edit action**

You have successfully configured the action with the provider.  
Edit the action or choose a different action category to configure a completely different action.

**Action category\*** Test  
Configure how your application is tested.

**Test actions**  
Choose from a list of test actions.

**Action name\*** Blazemeter

**Test provider\*** BlazeMeter

**BlazeMeter**  
Add BlazeMeter tests to your pipeline. Choose Connect to log into your BlazeMeter account and configure access for AWS CodePipeline.  
[Reconnect](#)

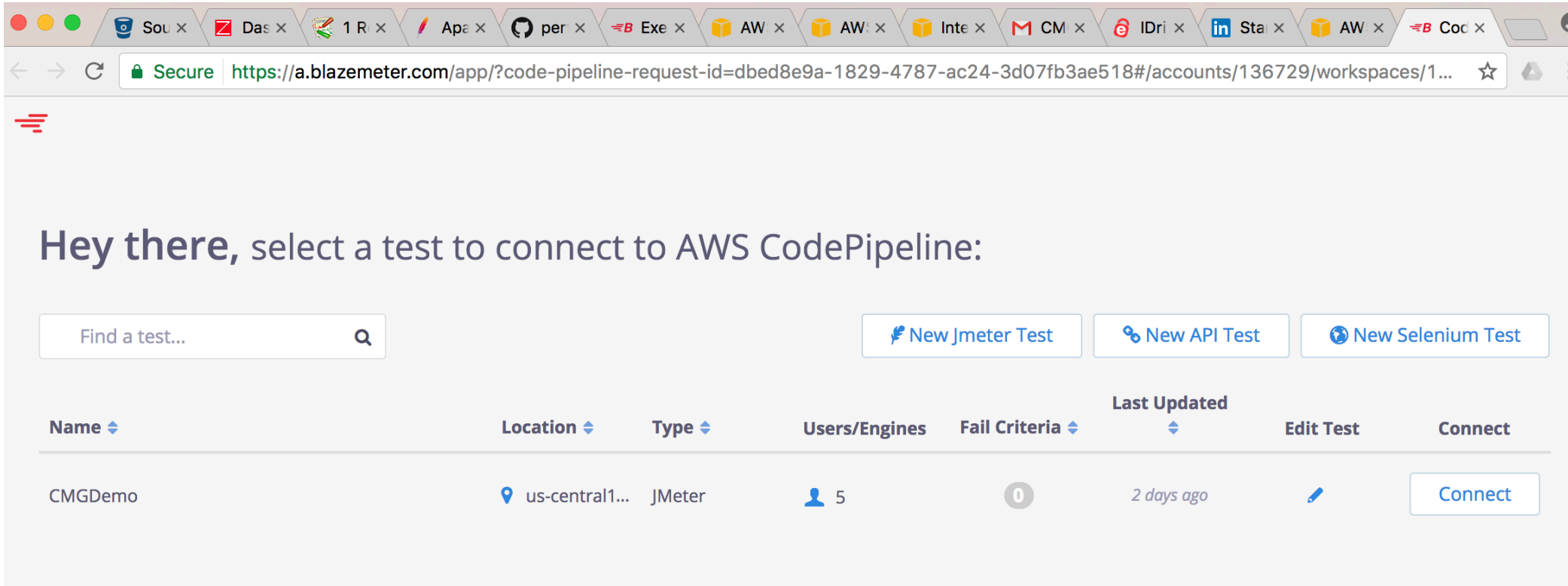
**Action configuration**  
TestId 5589982

**Input artifacts**


\* Required [Cancel](#) [Update](#)

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)










# ***TEST STAGE*** IN AWS CODEPIPELINE



Hey there, select a test to connect to AWS CodePipeline:

Find a test... 

[New Jmeter Test](#) [New API Test](#) [New Selenium Test](#)

Name 	Location 	Type 	Users/Engines	Fail Criteria 	Last Updated 	Edit Test	Connect
CMGDemo	 us-central1...	JMeter	 5	 0	2 days ago		<a href="#">Connect</a>