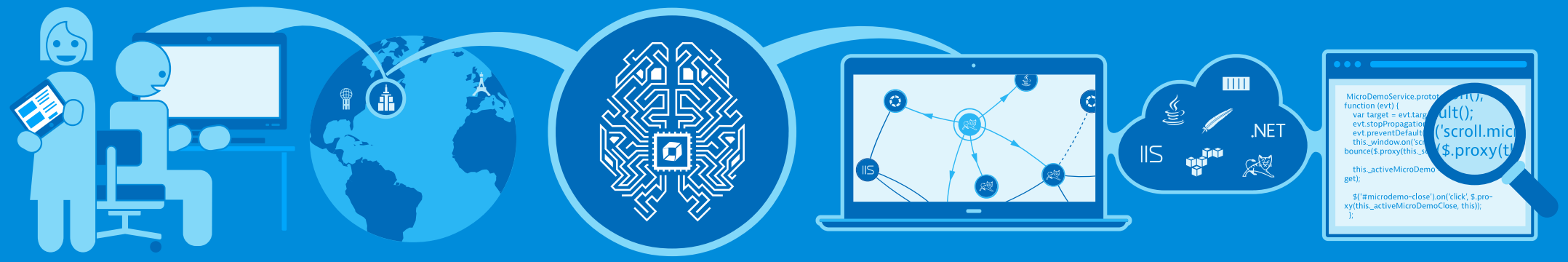
AWS “Hands-On” with Dynatrace

26 Oct 2017



Agenda

Steps of the exercise

- Sign up for Dynatrace free trial (if you do not have one)
- Hands on #1
  Setup Dynatrace AWS Monitoring Integration
- Hands on #2
  Deploy and Monitor NodeJS Beanstalk Application
- Hands on #3
  Deploying EC2 instance with a OneAgent

Pre-Requisits

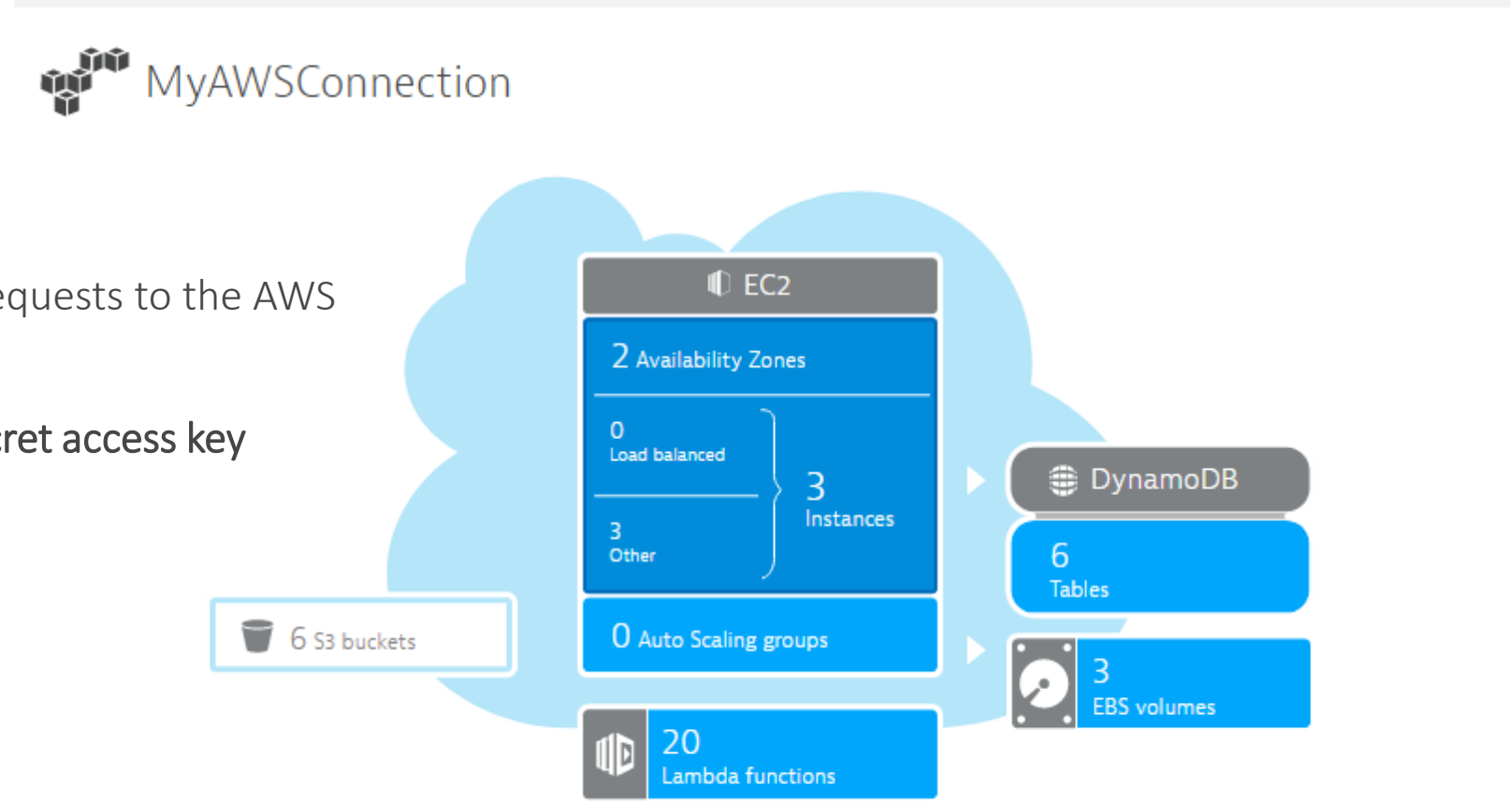
- Dynatrace SaaS or Managed Account
- AWS Account with administrative privileges
- For Troubleshooting
  - Create a Key Pair for EC2.
  - Download your key.pem
  - More instructions to Connect to EC2 via Linux (SSH), Windows (PuTTy): https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html

AWS Hands On 1

Goals: Setup Dynatrace AWS Monitoring Integration

Goal: See AWS CloudWatch Data in Dynatrace

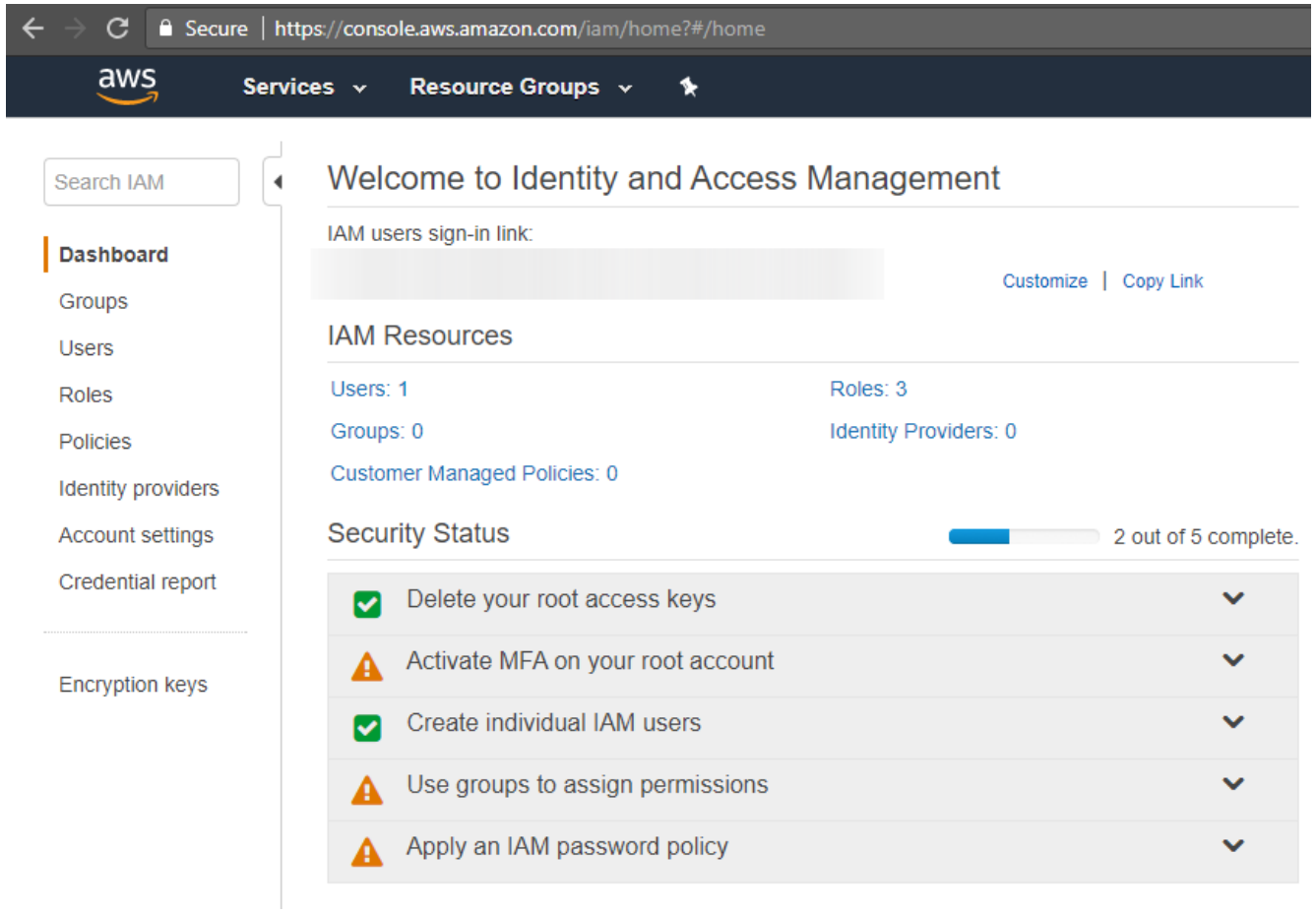
- Approach: Access Cloud Watch
  - via Access Key-based Authentication or
  - via Role-Based Authentication



- Key-based Authentication
  - Secure REST or Query protocol requests to the AWS service API
  - Generate **Access key ID** and a **Secret access key**

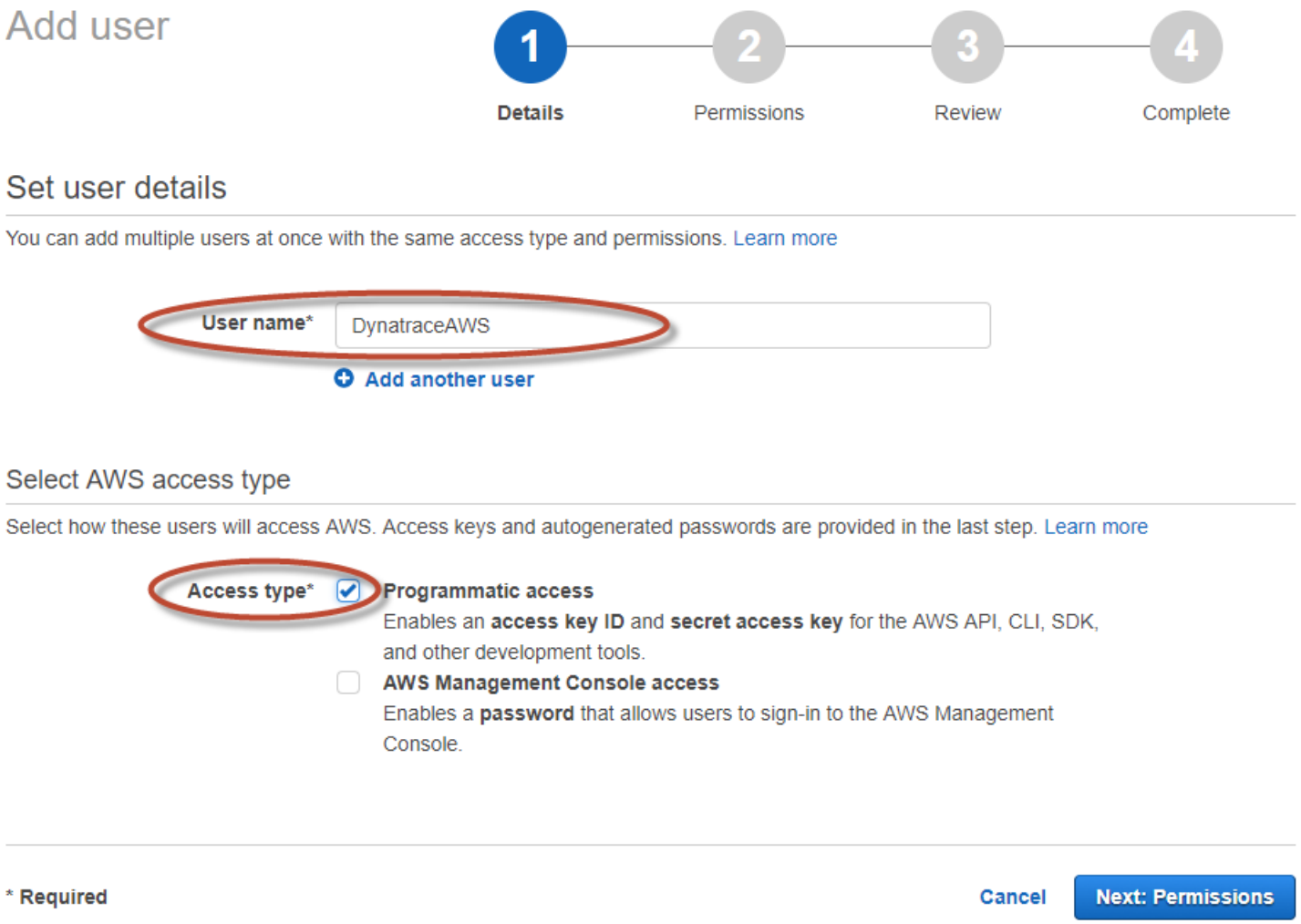
Step 1 – Create a user in IAM

- Go to Identity and Access Management (IAM) in your Amazon Console

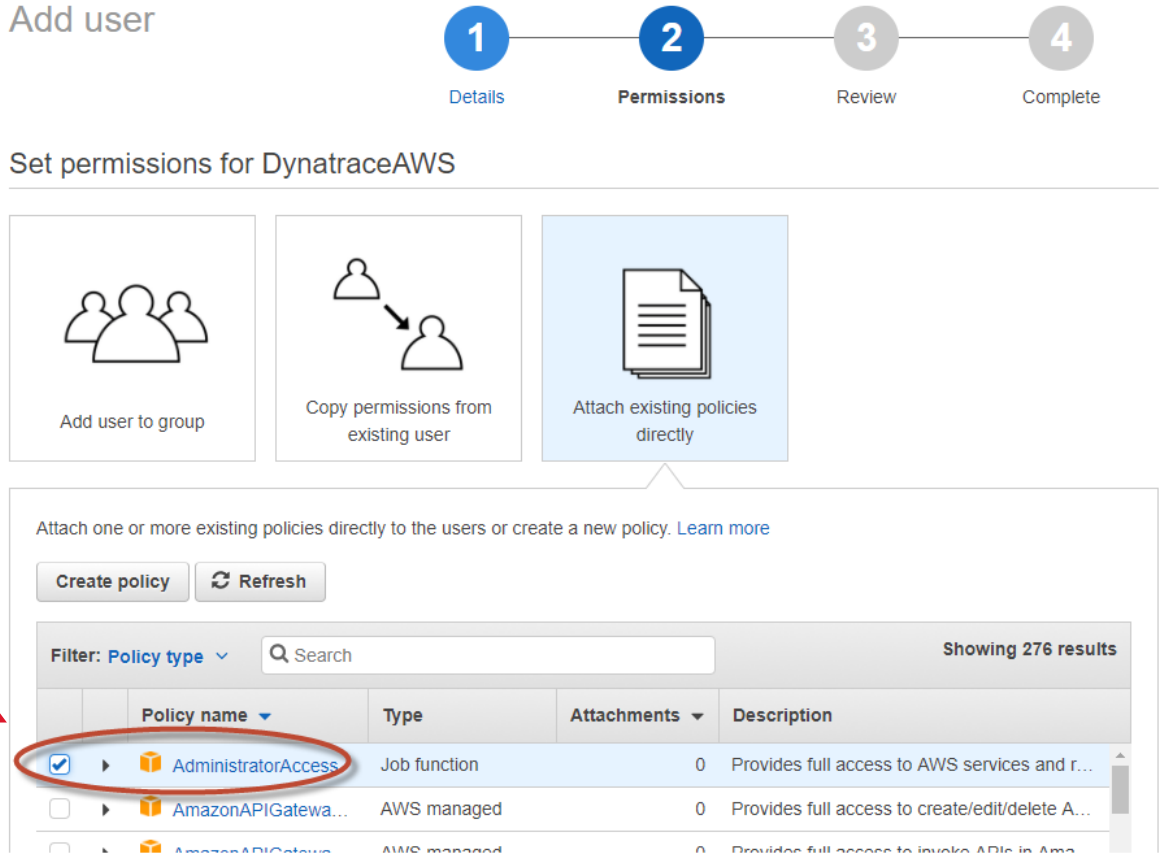


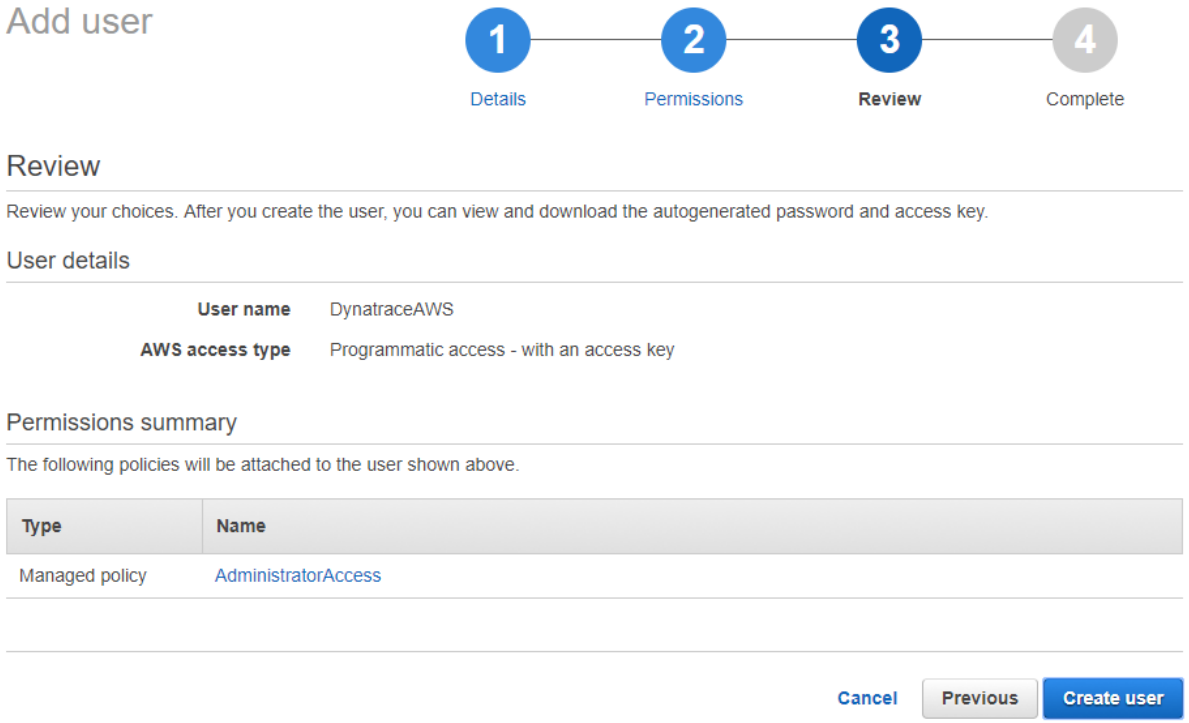
Step 2

- Go to **Users** and click **Add Users**
- Enter a User name
- Select "**Programmatic access**" for Access type
- Click on Next



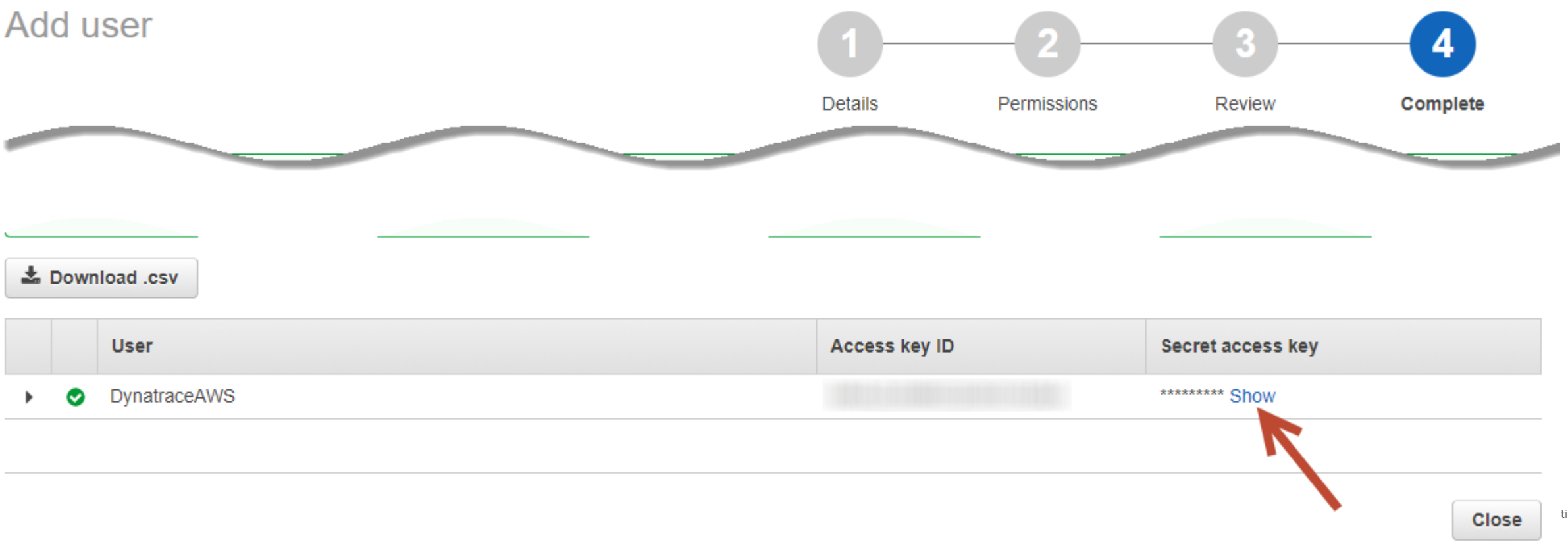
- Set Permissions
- Check **AdministratorAccess**
- Click Next to Review
- Click **Create User**





Step 3

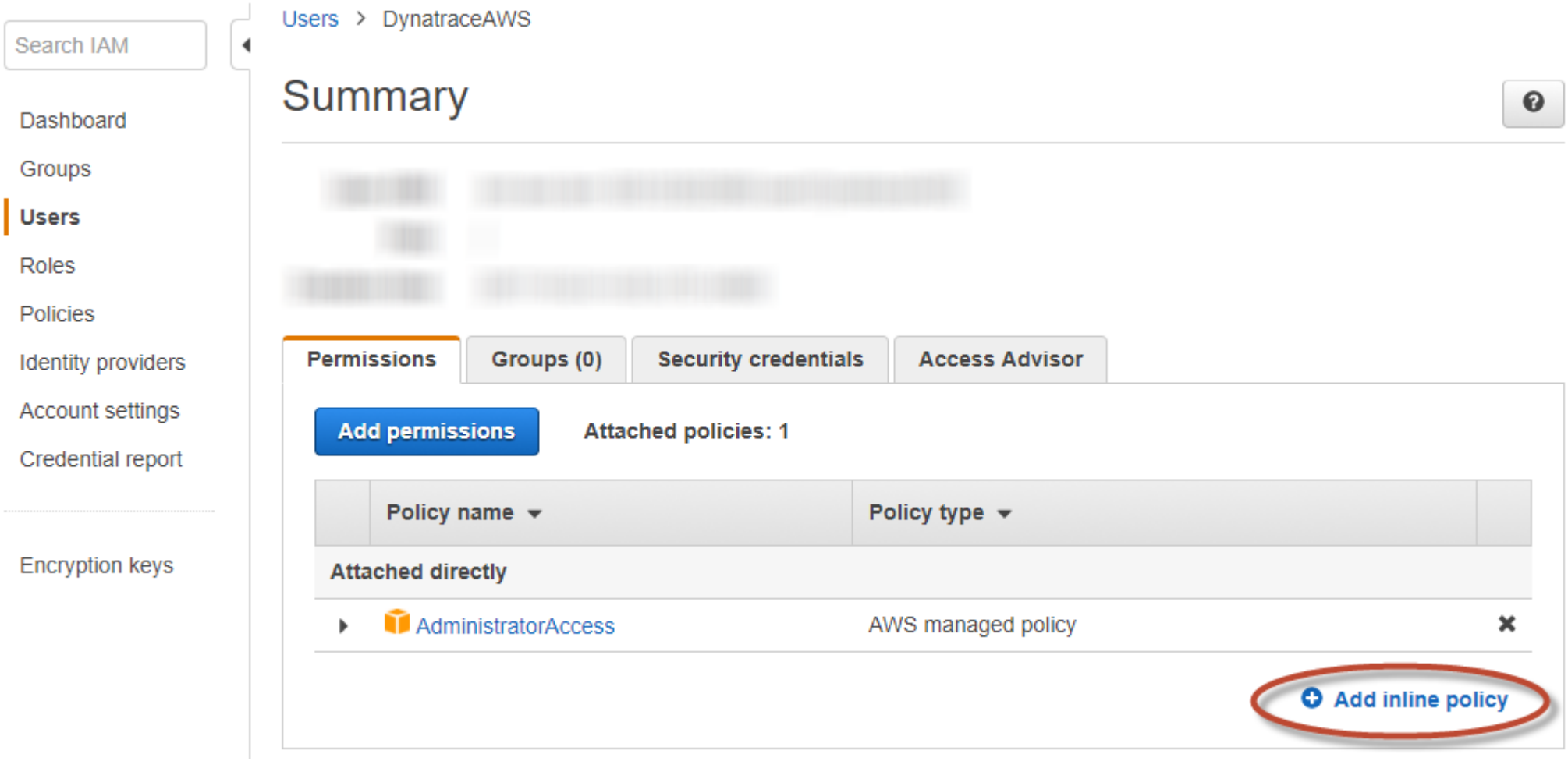
- Copy out the values of Access Key ID and Secret access key
- Alternatively, you can download the .csv file and copy it from there



tial

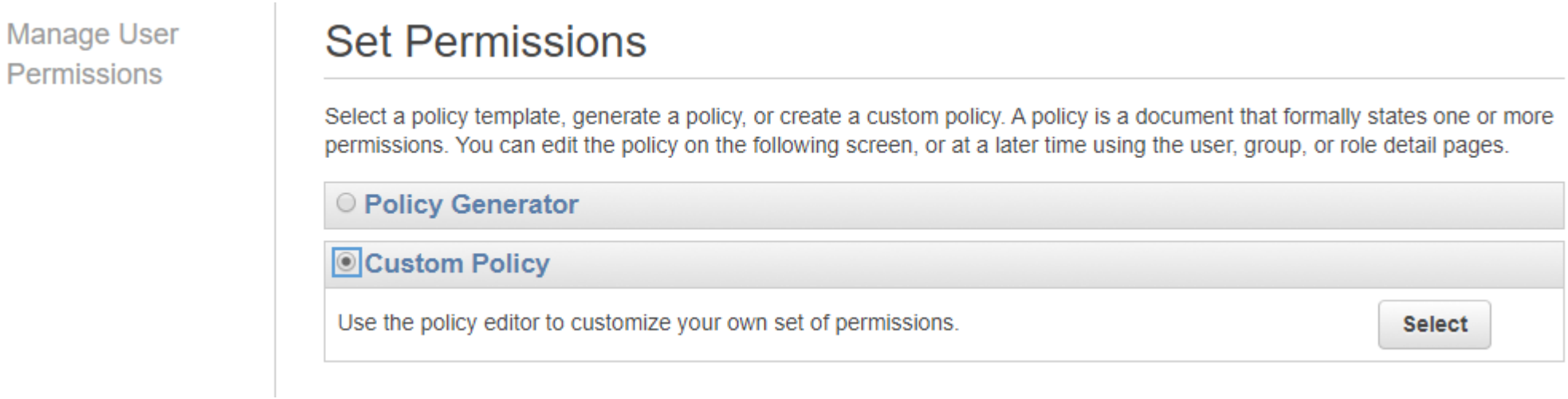
Step 4

- In **Permissions** expand **+Add Inline Policies**



Step 5

- In the **Set Permissions** section, select **Custom Policy**
- Create the policy by clicking on **Select**



Step 6

- In the **Policy Name field**, type a name for the policy (for example DynatraceAWS-policy)
- In the **Policy Document** field, copy out the contents from aws-policy.json and paste it
- Click on **Apply Policy**

Manage User Permissions

Review Policy

Customize permissions by editing the following policy document. For more information about the access policy language, see Overview of Policies in the *Using IAM* guide. To test the effects of this policy before applying your changes, use the IAM Policy Simulator.

Policy Name

DynatraceAWS-policy

Policy Document

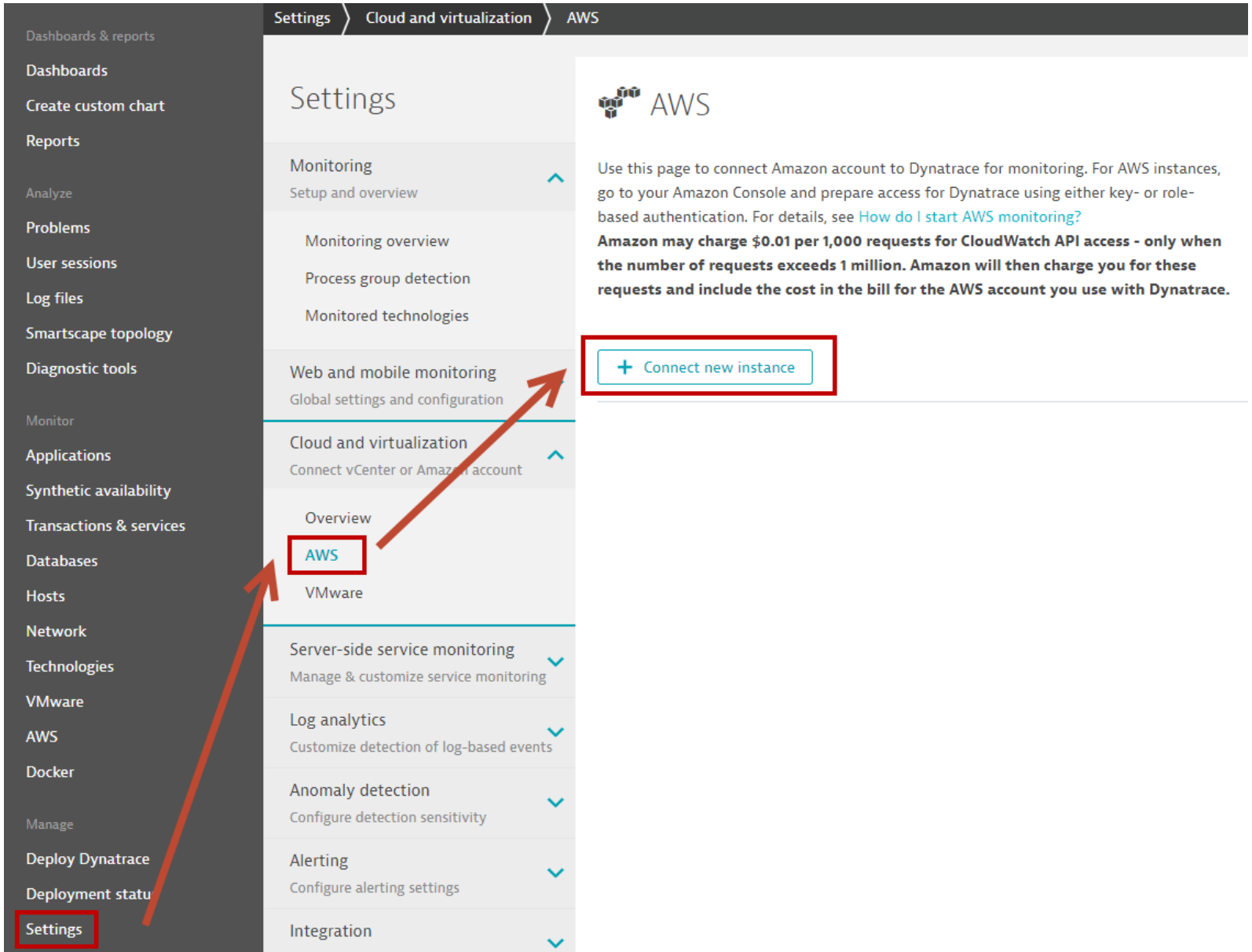
```
1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Effect": "Allow",
6        "Action": [
7          "autoscaling:Describe*",
8          "cloudwatch:Describe*",
9          "cloudwatch:Get*",
10         "cloudwatch:List*",
11         "ec2:Describe*",
12         "elasticloadbalancing:Describe*",
13         "rds:DescribeDBInstances",
14         "rds:DescribeEvents",
15         "rds:List*",
16         "dynamodb:DescribeTable",
17         "dynamodb:ListTables",
18         "lambda:ListFunctions",
19         "lambda:GetFunction",
20         "elasticbeanstalk:DescribeEnvironments",
21         "elasticbeanstalk:DescribeEnvironmentResources",
22         "s3:ListAllMyBuckets",
23         "sts:GetCallerIdentity"
```

☑ Use autoformatting for policy editing

Cancel | Validate Policy | Apply Policy

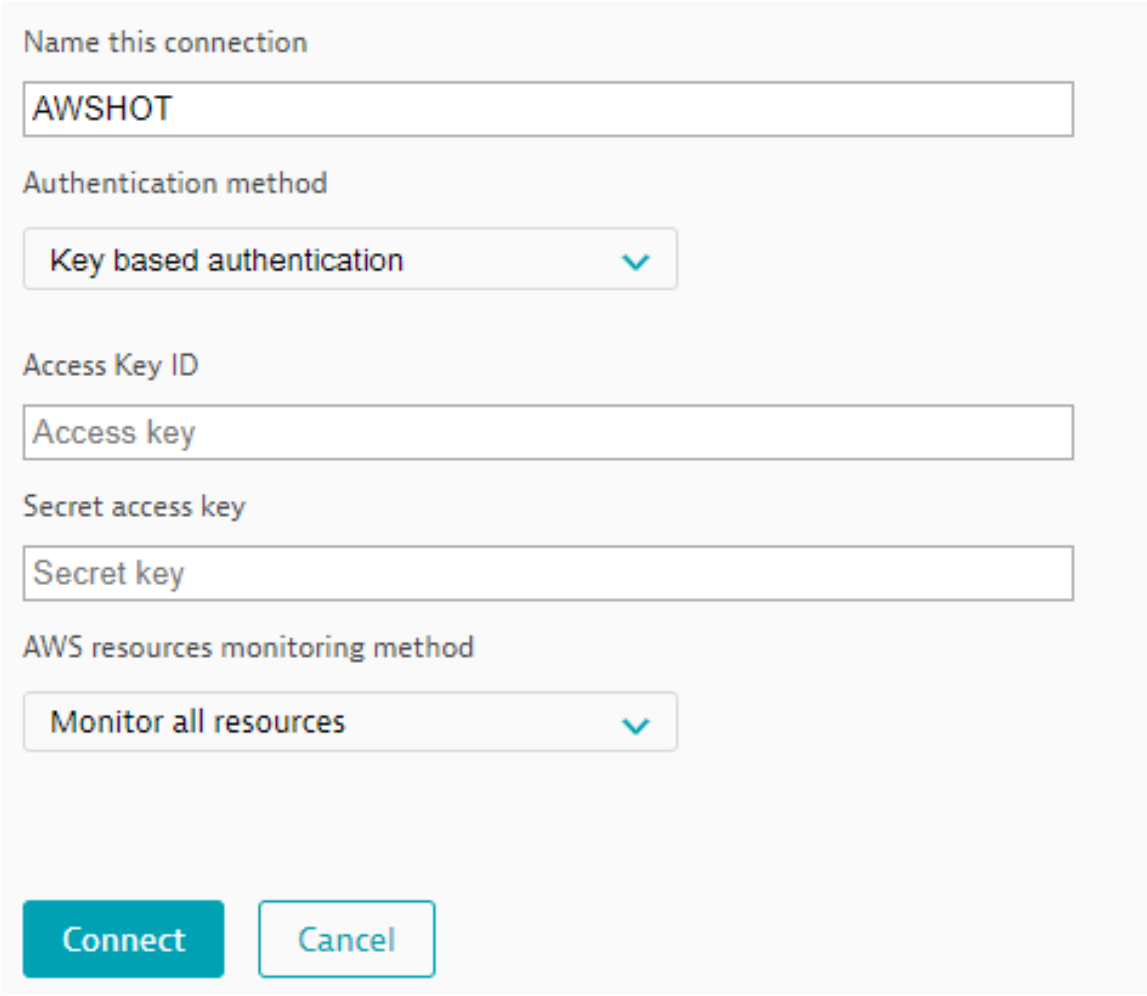
Step 7 – Connect your Amazon account to Dynatrace

- Go to Settings > Cloud & virtualization > AWS & VMware and click Connect new instance

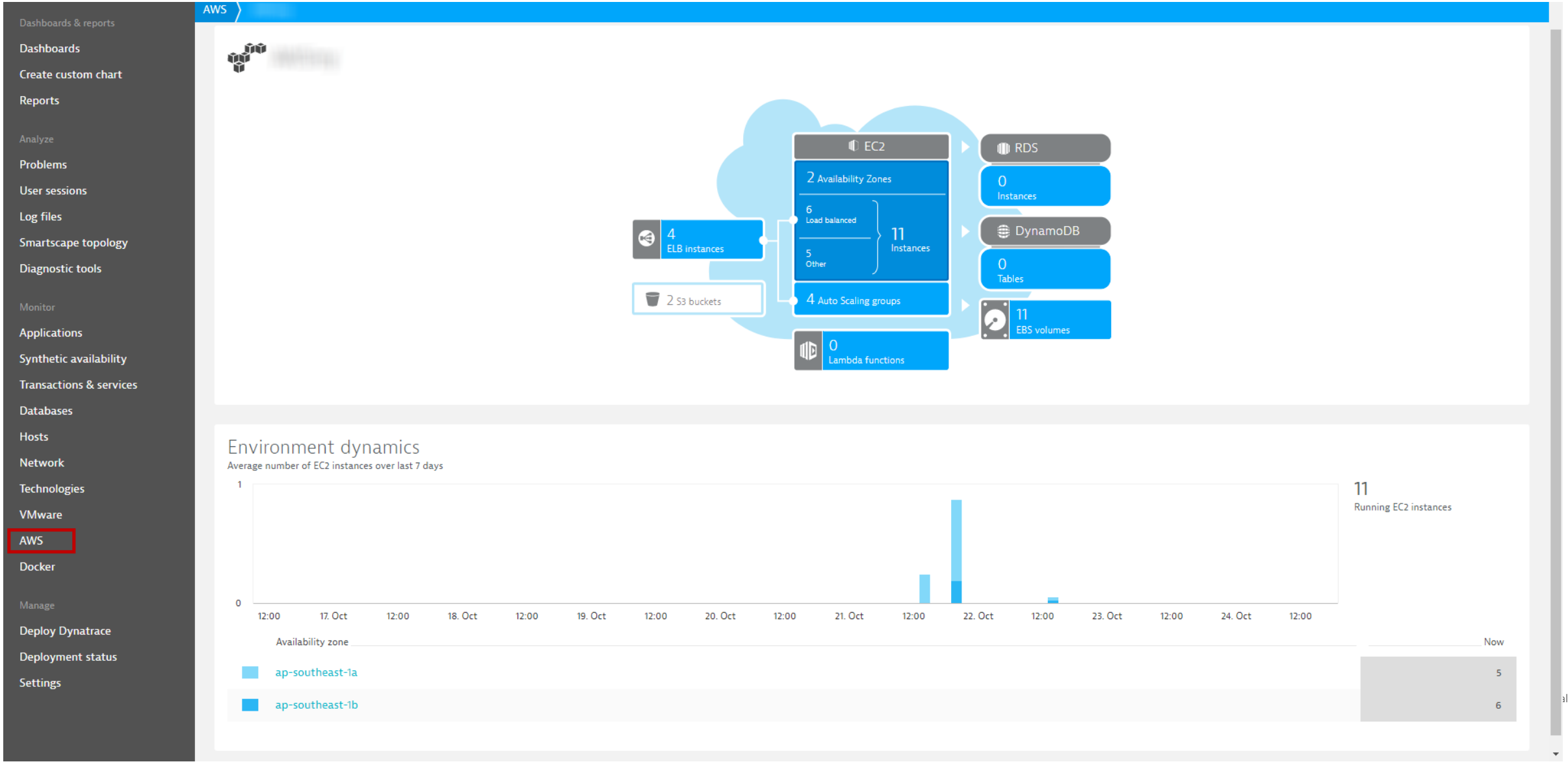


Step 8

- Create a name for this connection. This is mandatory. Dynatrace needs this name to identify and display the connection.
- In the Access key ID field, paste the key you created in Amazon for Dynatrace access.
- In the Secret access key field, paste the key you created in Amazon for Dynatrace access.
- Click Connect to verify and save the connection.



End result


Dashboards & reports
Dashboards
Create custom chart
Reports
Analyze
Problems
User sessions
Log files
Smartscape topology
Diagnostic tools
Monitor
Applications
Synthetic availability
Transactions & services
Databases
Hosts
Network
Technologies
VMware
AWS
Docker
Manage
Deploy Dynatrace
Deployment status
Settings
AWS
EC2
2 Availability Zones
6 Load balanced
5 Other
11 Instances
4 Auto Scaling groups
4 ELB instances
2 S3 buckets
0 Lambda functions
RDS
0 Instances
DynamoDB
0 Tables
11 EBS volumes
Environment dynamics
Average number of EC2 instances over last 7 days
11
Running EC2 instances
Availability zone
Now
ap-southeast-1a
5
ap-southeast-1b
6

Cloud Delivery Models

Traditional on Premise	Infrastructure-as-a-Service	Platform-as-a-Service	Software-as-a-Service
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
OS	OS	OS	OS
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

Client Manages

Vendor Manages in Cloud

Standardization – Lower Costs –Faster Time to Value

AWS Hands On 2

Goals: Deploy and Monitor NodeJS Beanstalk Application

101 on BeanStalk

- What is Beanstalk?
  - AWS Hosts and Scales Application Servers (Node.JS, Java, …) for you
  - Developer only provides ZIP or WAR file
  - ZIP/WAR File gets deployed on Application Server during startup
  - Automatically Load Balanced / Auto-Scaling …
- Beanstalk ZIP File Structure

Name	Date modified
.ebextensions	26.06.2017 14:04
app.js	27.06.2017 17:04
cron.yaml	17.02.2015 23:37
index.html	27.06.2017 17:23
package.json	17.02.2015 23:37

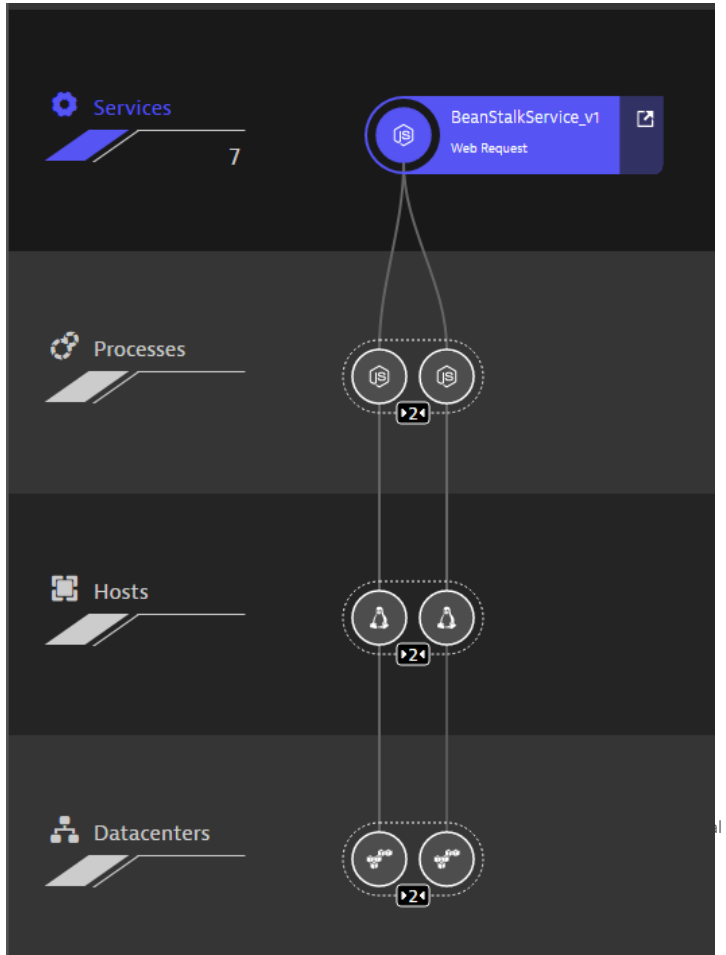
Name	Date modified
dynatrace.config	26.06.2017 14:28
logging.config	17.02.2015 23:37
version.config	27.06.2017 00:56

```
---
option_settings:
  - option_name: DYNATRACE_ONEAGENT_DOWNLOAD
    value: YOURFULLONEAGENTDOWNLOADURL

files:
  /opt/elasticbeanstalk/hooks/appdeploy/pre/01_dynatrace_install.sh:
    content: |
        DYNATRACE_ONEAGENT_DOWNLOAD=$(/opt/elasticbeanstalk/bin/get-c
        if [ \"x$DYNATRACE_ONEAGENT_DOWNLOAD\" == \"x\" ]
        then
          DYNATRACE_ONEAGENT_DOWNLOAD=$(/opt/elasticbeanstalk/bin/get
        fi
```

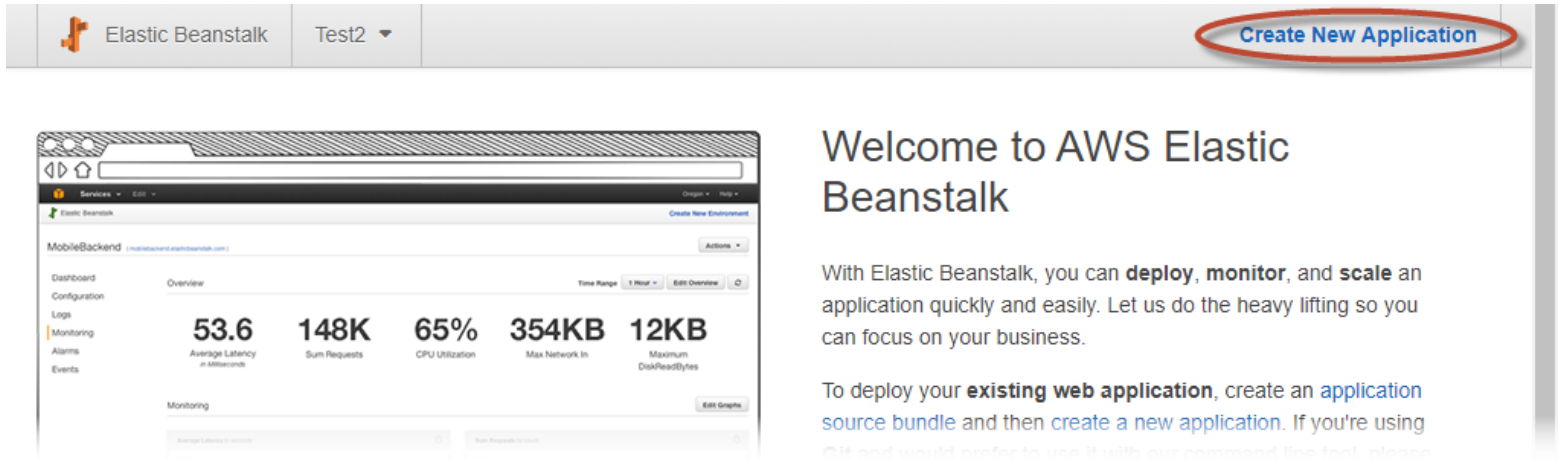
Goal: See Full-Stack Tracing of Node.JS Beanstalk Application

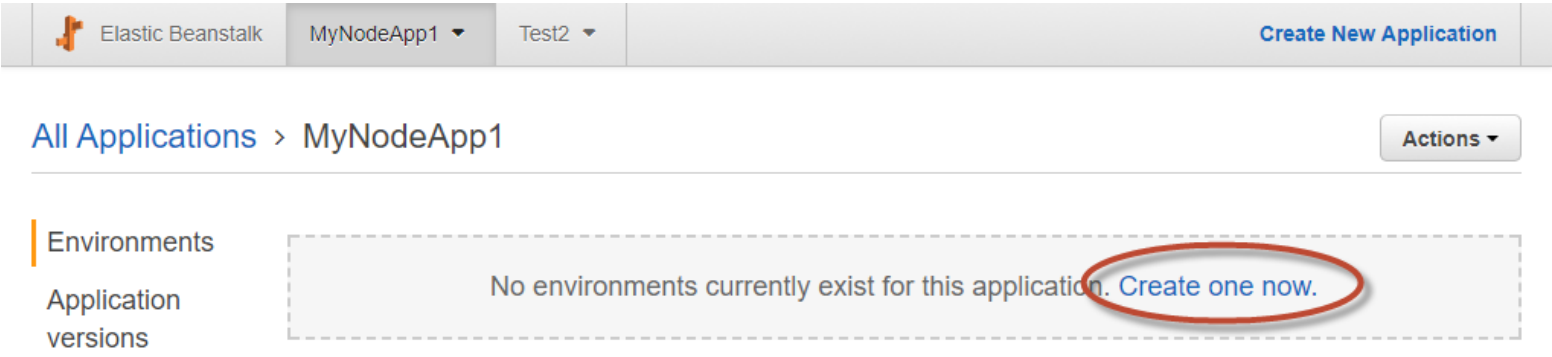
- Using .ebextension mechanism to
  - Execute Agent Download and Install during EC2 Launch
  - Pass additional tags, e.g: BeanStalkVersion_v2
- IMPORTANT
  - Mac: make sure .zip file contains .ebextension folder (any file with a "." is hidden)
  - Mac & Win: Make sure package.json is at the root of the .zip file, instead of nested in a folder
- Additional Hands-On Use Cases
  - Load Balancing / Auto-Scaling BeanStalk Service
  - RUM Configuration: User Tagging and jQuery Support



Step 1 Deploy the sample App

- Logon to AWS and Navigate to Elastic Beanstalk
- Create a new application
- Give it a name. Then create an environment.
- Choose **Web Server environment**





- Select **Node.js** as the platform
- Upload NodeJSBeanStalkSample.zip
- Click on Configure more options

Base configuration

Tier Web Server (Choose tier)

Platform Preconfigured platform

Platforms published and maintained by AWS Elastic Beanstalk.

Node.js

Custom platform NEW

Platforms created and owned by you. Learn more

-- Choose a custom platform --

Application code Sample application

Get started right away with sample code.

Existing version

Application versions that you have uploaded for **MyNodeApp1**.

-- Choose a version --

Upload your code

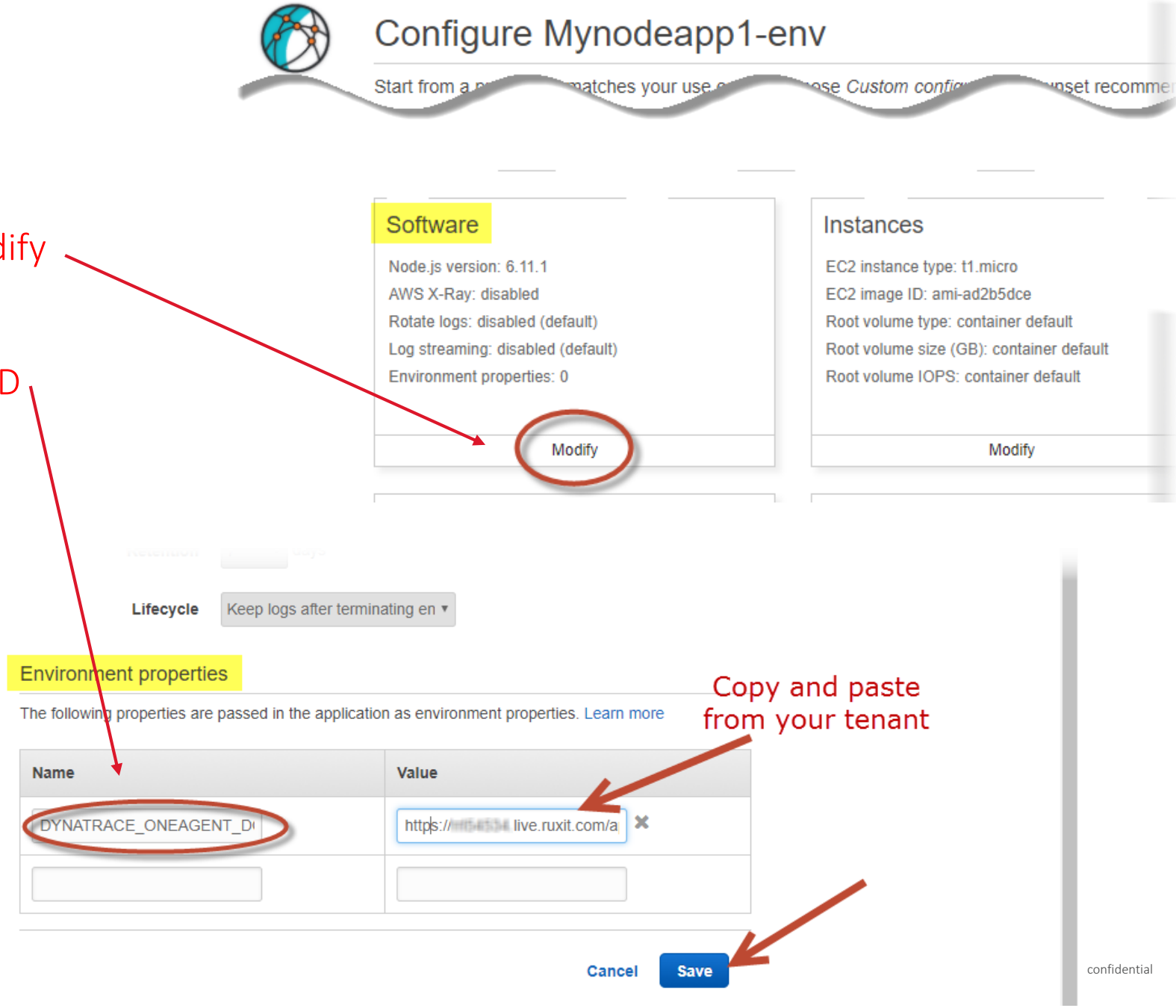
Upload a source bundle from your computer or copy one from Amazon S3.

Upload ZIP or WAR

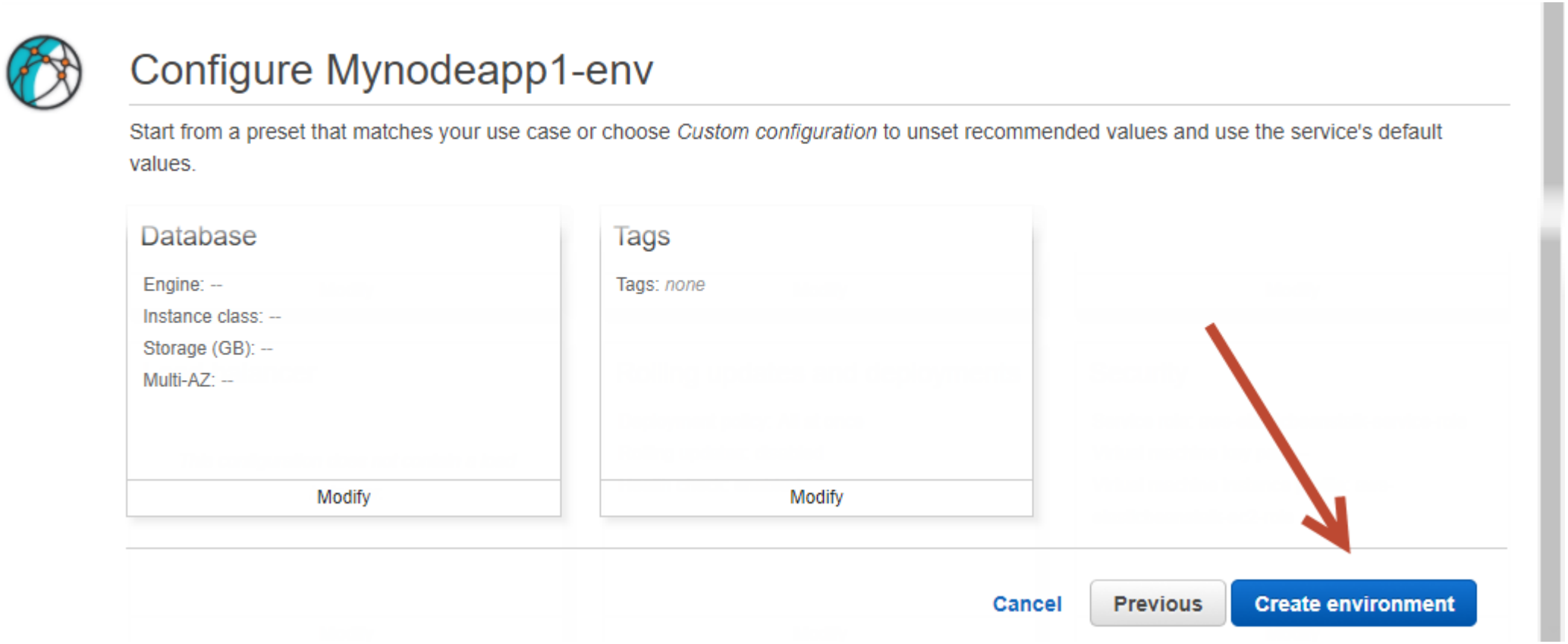
Cancel | Configure more options | Create environment

Step 2

- Under Software section, click on Modify
- Add the exact text DYNATRACE_ONEAGENT_DOWNLOAD
- And copy the full download URL for your OneAgent from you own tenant
- Click on Save



- Click on Create Environment



All Applications › MyNodeApp1 › Mynodeapp1-env (Environment ID: [redacted])

Actions

Creating Mynodeapp1-env

This will take a few minutes.

7:20pm Environment health has transitioned to Pending. Initialization in progress (running for 10 seconds). There are no instances.

7:20pm Created EIP: [redacted]

7:20pm Created security group named: [redacted]

7:19pm Using elasticbeanstalk-ap-southeast-1-603123023980 as Amazon S3 storage bucket for environment data.

7:19pm createEnvironment is starting.

Learn More

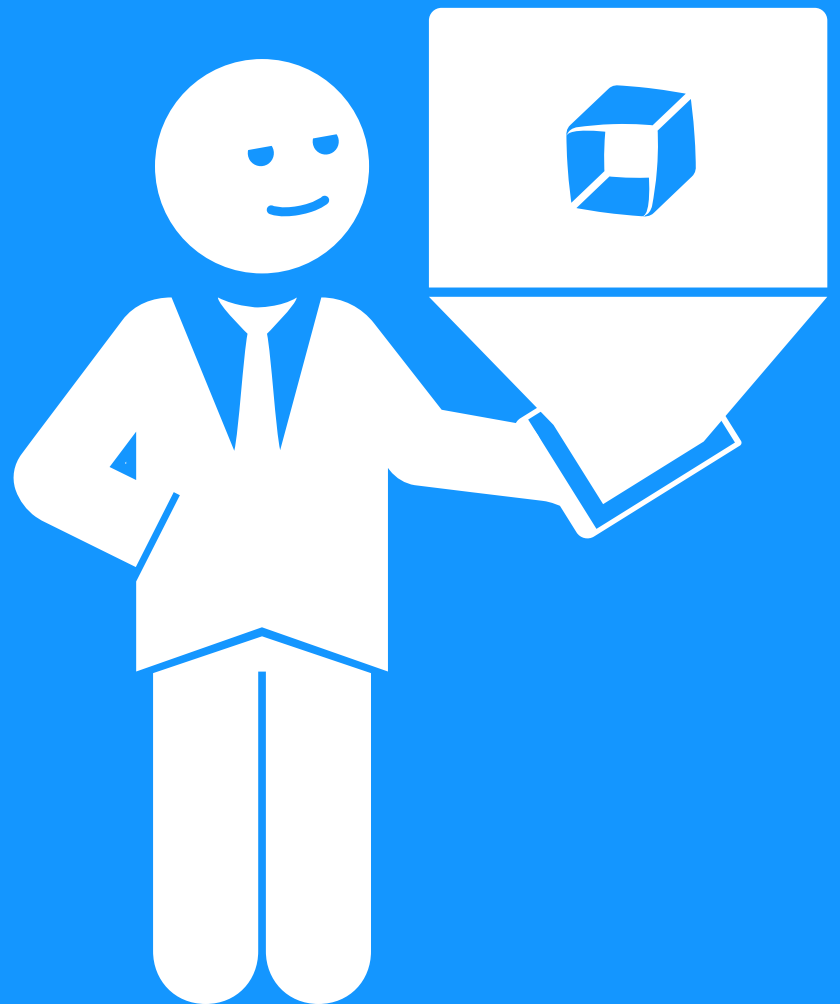
Get started using Elastic Beanstalk
Modify the code
Create and connect to a database
Add a custom domain

Featured

Create your own custom platform

Command Line Interface (v3)

Installing the AWS EB CLI
EB CLI Command Reference



Explore Dynatrace!

Elastic Beanstalk | MyNodeApp1 | Test2 | Create New Application

All Applications > MyNodeApp1 > Mynodeapp1-env (Environment ID: [redacted], URL: Mynodeapp1-env.eq7n9ppeaj.ap-southeast-1.elasticbeanstalk.com) Actions

Creating Mynodeapp1-env

This will take a few minutes....

7:25pm Successfully launched environment: Mynodeapp1-env

7:25pm Environment health has transitioned from Pending to Ok. Initialization completed 28 seconds ago and took 5 minutes.

7:21pm Waiting for EC2 instances to launch. This may take a few minutes.

7:21pm Added instance [redacted] to your environment.

7:20pm Environment health has transitioned to Pending. Initialization in progress (running for 10 seconds). There are no instances.

7:20pm Created EIP: [redacted]

7:20pm Created security group named: [redacted]

7:19pm Using elasticbeanstalk-ap-southeast-1-603123023980 as Amazon S3 storage bucket for environment data.

7:19pm createEnvironment is starting.

Learn More

Get started using Elastic Beanstalk
Modify the code
Create and connect to a database
Add a custom domain

Featured

Create your own custom platform

Command Line Interface (v3)

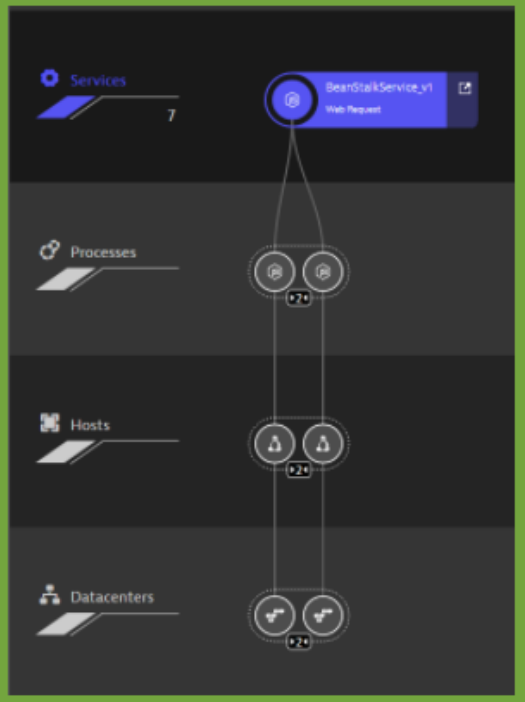
Installing the AWS EB CLI
EB CLI Command Reference

Feedback | English (US)

 Privacy Policy | Terms of Use

mynodeapp1-env.eq7n9ppeaj.ap-southeast-1.elasticbeanstalk.com

Congratulations



This might be your first AWS Beanstalk App you ever deployed. The code base of this application was taken from the AWS Elastic Beanstalk Tutorial but was slightly modified for the Dynatrace AWS Tutorial that you can find here! The goal of this tutorial is to enable Full Stack Monitoring with Dynatrace on your Beanstalk Application. Following is a screenshot of Dynatrace Smartcape in case you deploy this application in a scalability group: If you want to learn

Lets trace some code!

Sleep Setting (in ms): 1000

Say Something : Just say whatever you want Echo

Invoke Server Side URL (full URL please) : https://www.amazon.com Invoke

Your Username (can be used for user tagging) : Joe Mueller
LOG ME IN :-)

Get MyVersion!

Request to 'https://www.amazon.com' returned with HTTP Status: 200 and response body length: 448779

External Links: To learn more about Beanstalk

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploy an Express Application to AWS Elastic Beanstalk
- Deploy an Express Application with Amazon ElastiCache to AWS Elastic Beanstalk
- Deploy a Geddy Application with Amazon ElastiCache to AWS Elastic Beanstalk
- Customizing and Configuring a Node.js Container
- Working with Logs

Hosts

Search

All 1

State

Running 1

Monitoring mode

Full-stack 1

All hosts

1 host

Monitor another host

Filtered by:

Name	State	Host units	CPU usage	Memory usage	Disk latency	Network traffic
Mynodeapp1-env	running	0.1	41 %	31 % of 590 MB	81 ms	371 kbit/s

Hosts > Mynodeapp1-env

Mynodeapp1-env

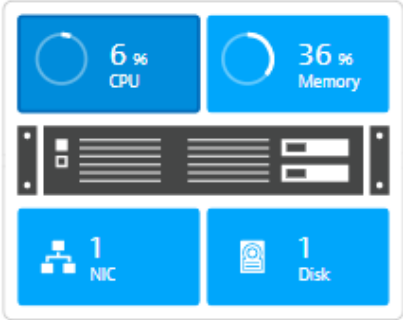
Uptime: 5 minutes

Edit ···

Properties and tags

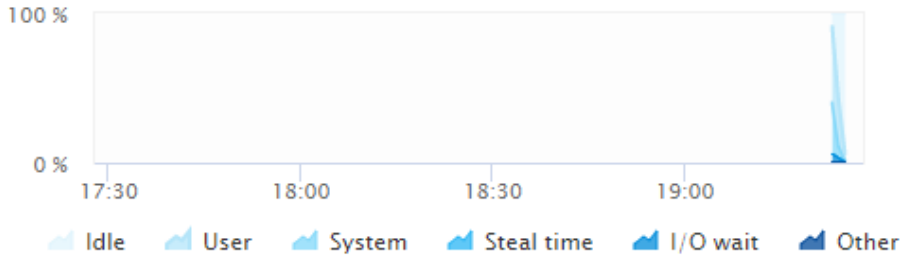
[AWS]Name: Mynodeapp1-env

more...



CPU usage 5.82 % | CPU AWS

Actively used CPU of the host as a percentage of available CPU.

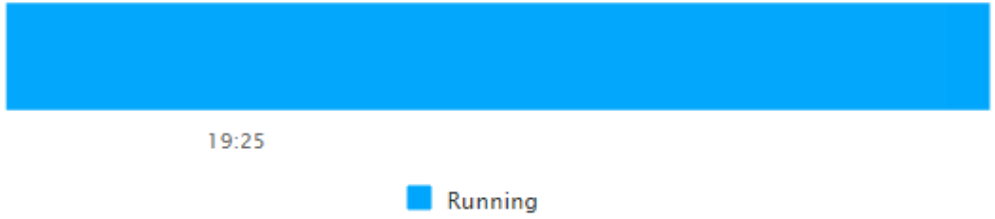


Consuming processes

No problems in last 72 hours

100% Availability

0 min total downtime



Processes

Ruby
command-processor
healthd

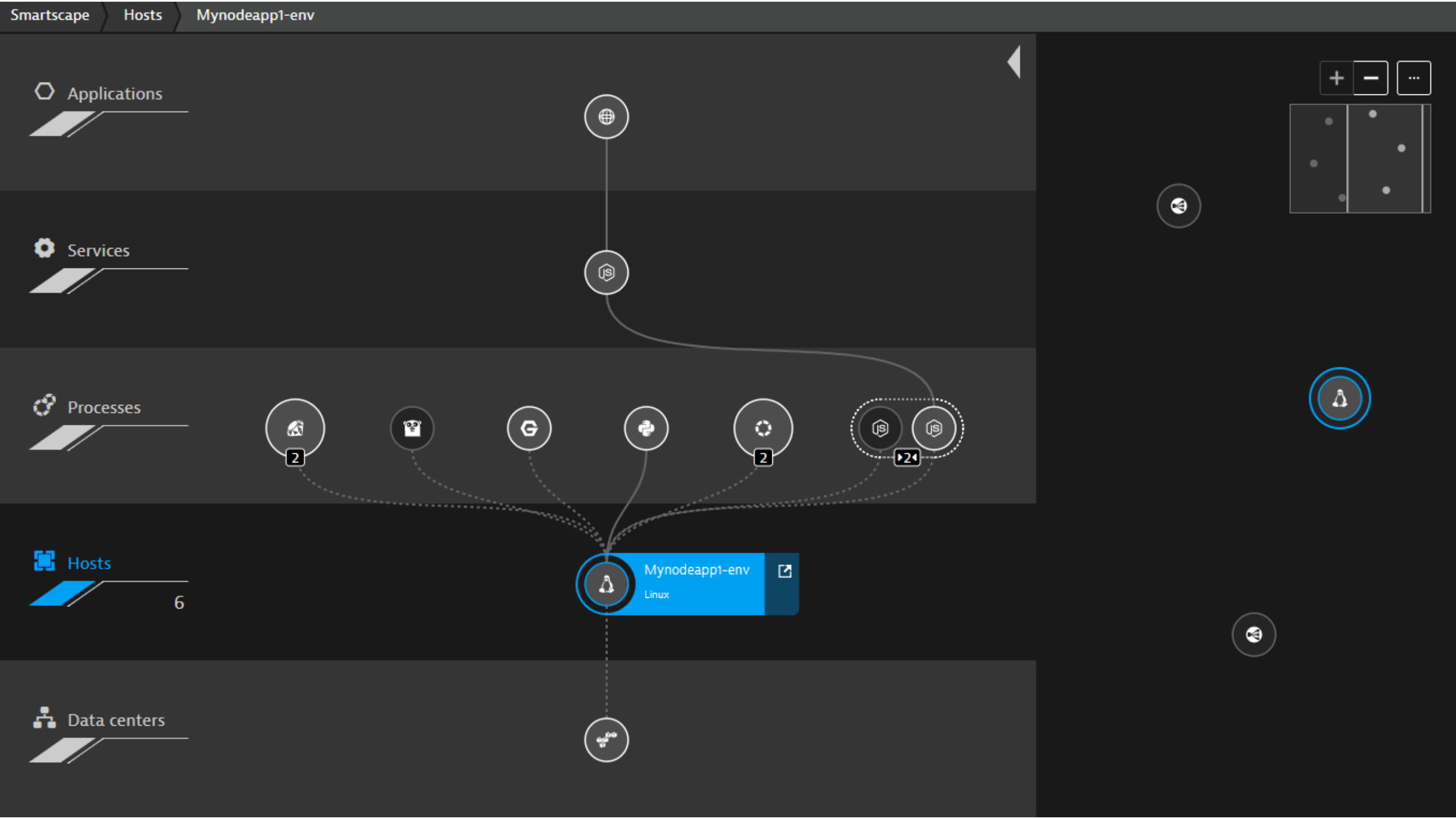
Nginx
nginx

Node.js
NodeBeanstalk_Version1
NodeBeanstalk_Version1

Go
xray

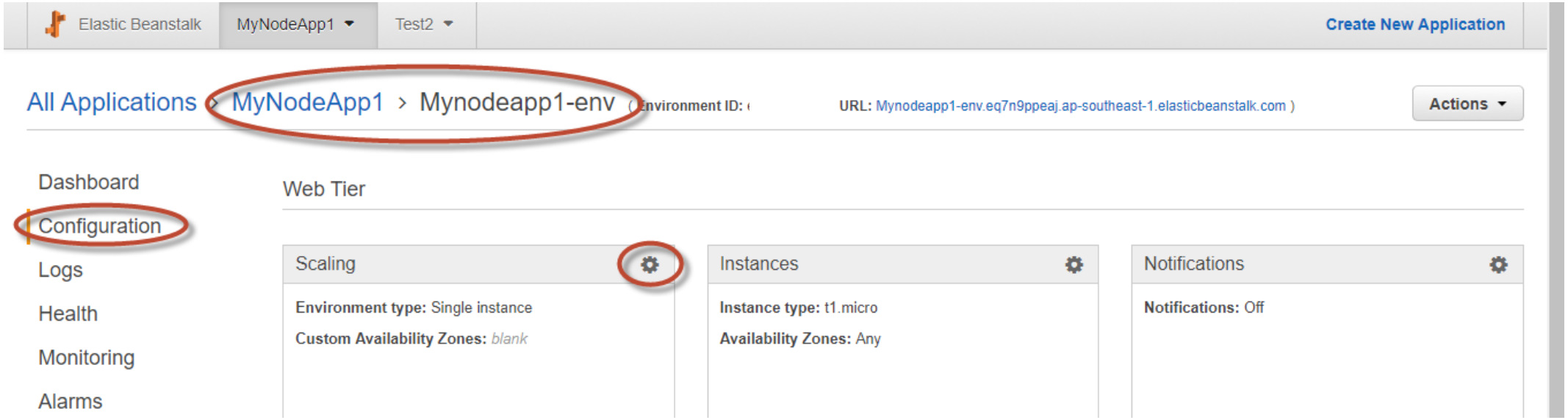
Restart 1 process to get full visibility

All processes


Smartscape
Hosts
Mynodeapp1-env
Applications
Services
Processes
Hosts
6
Data centers
Mynodeapp1-env
Linux

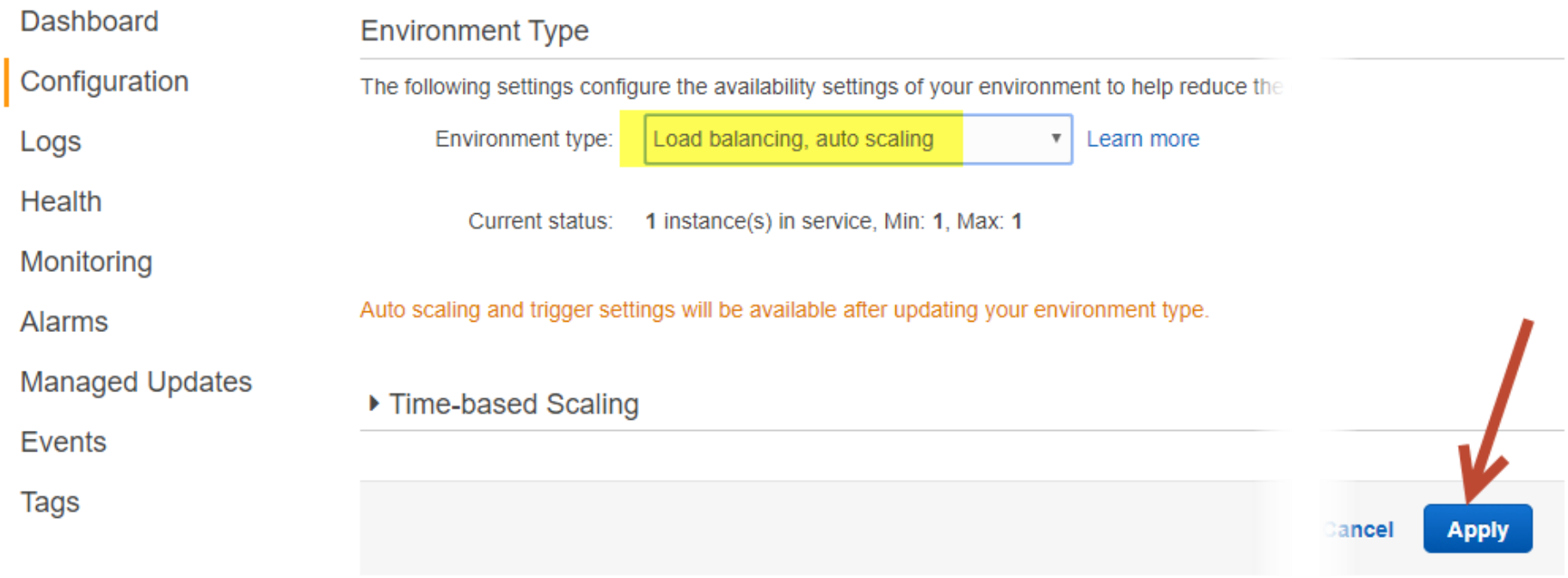
Additional step: Load Balancing

- Go to your Beanstalk Environment in AWS Console
- Click on Configuration – Scaling



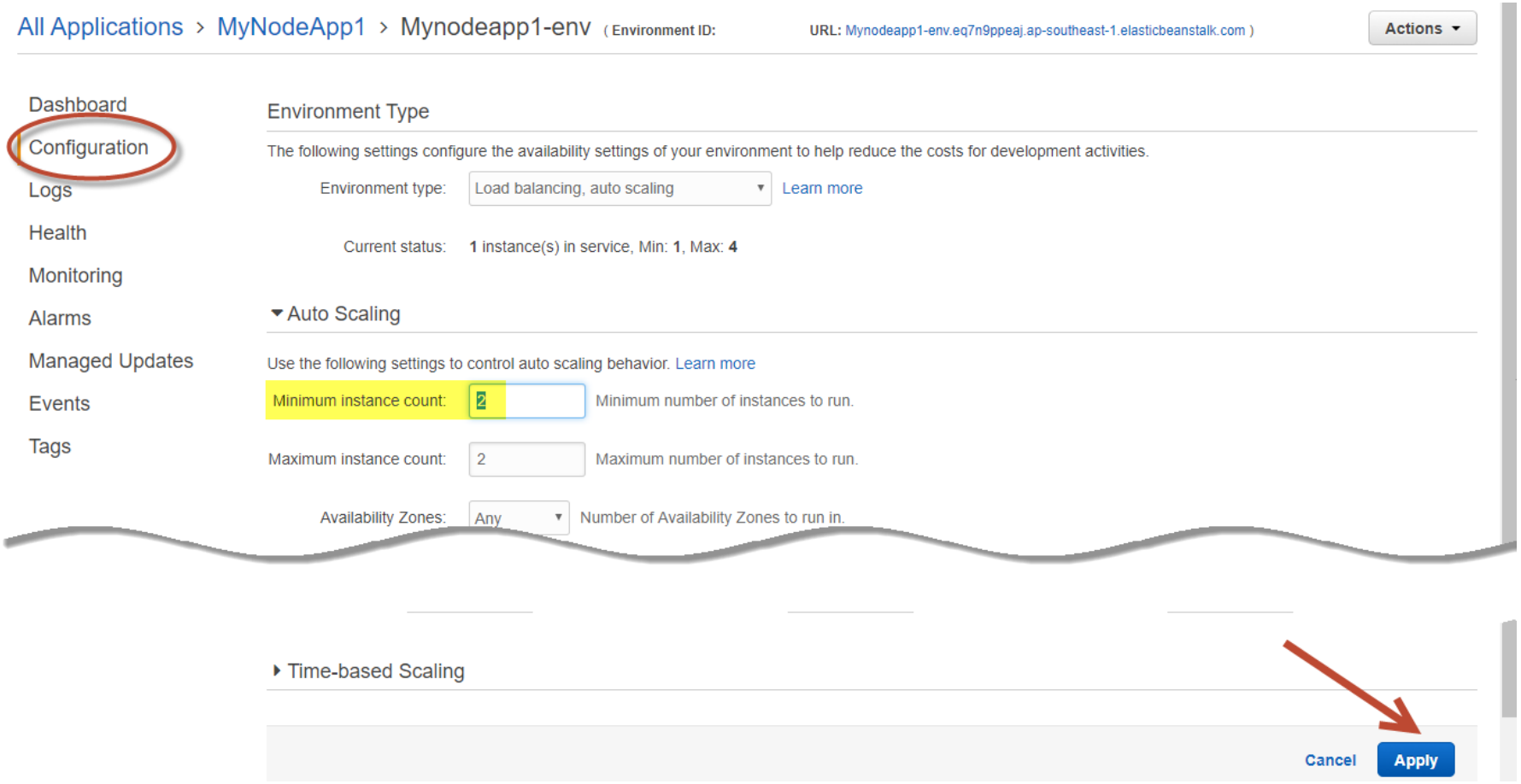
Additional step: Load Balancing

- Change the environment type to "Load balancing, auto scale"
- Apply the changes and let Beanstalk restart



Additional step: Load Balancing

- Go back to the same settings after restart and change auto scale to minimum instances of 2
- Apply changes and validate that Dynatrace detects both instances



All Applications › MyNodeApp1 › Mynodeapp1-env (Environment ID: URL: Mynodeapp1-env.eq7n9ppeaj.ap-southeast-1.elasticbeanstalk.com) Actions

- Dashboard
- Configuration
- Logs
- Health
- Monitoring
- Alarms
- Managed Updates
- Events
- Tags

Overview

Refresh

Health

Ok

Causes

Running Version

v1

Upload and Deploy

Configuration

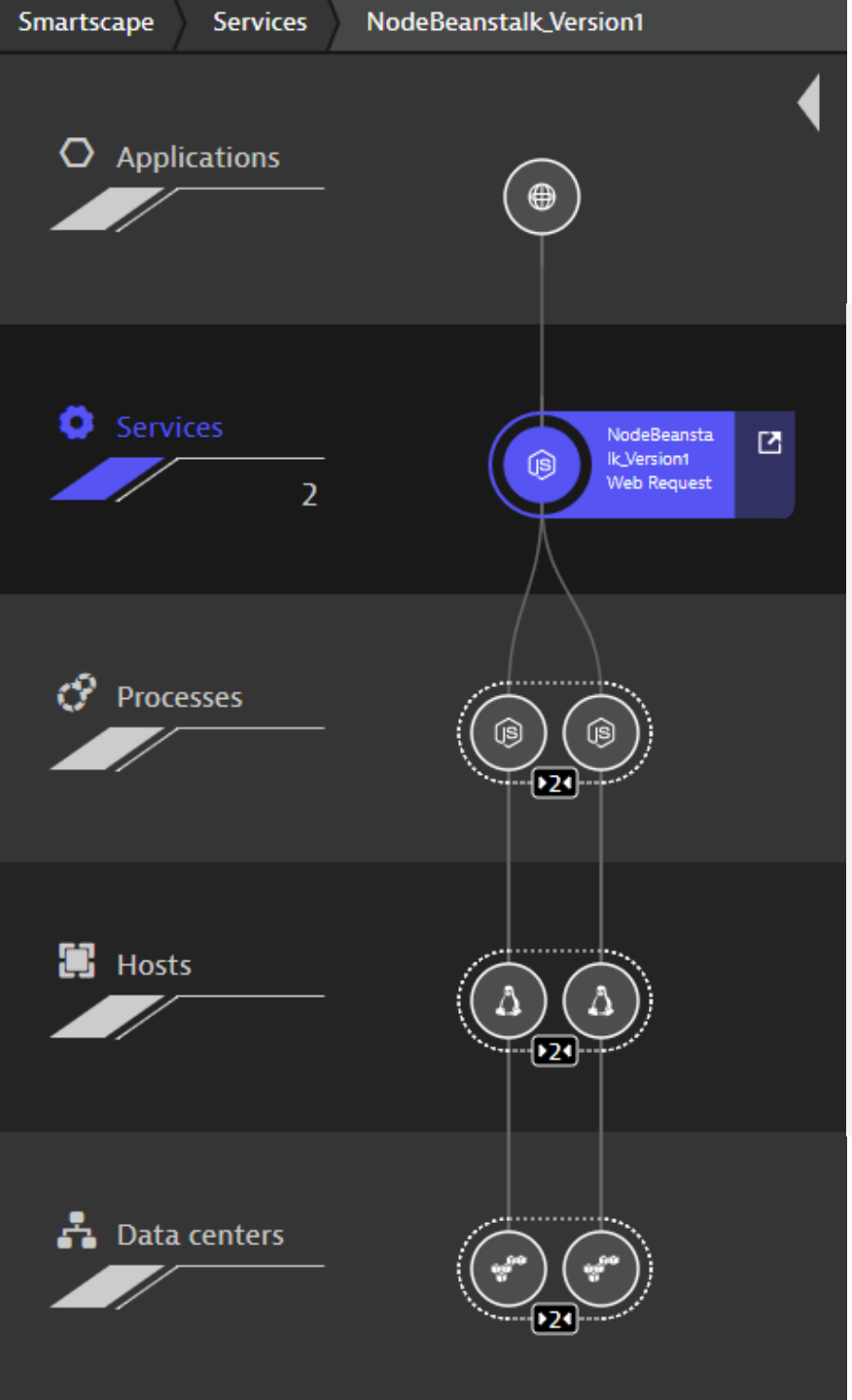
64bit Amazon Linux 2017.03
v4.3.0 running Node.js

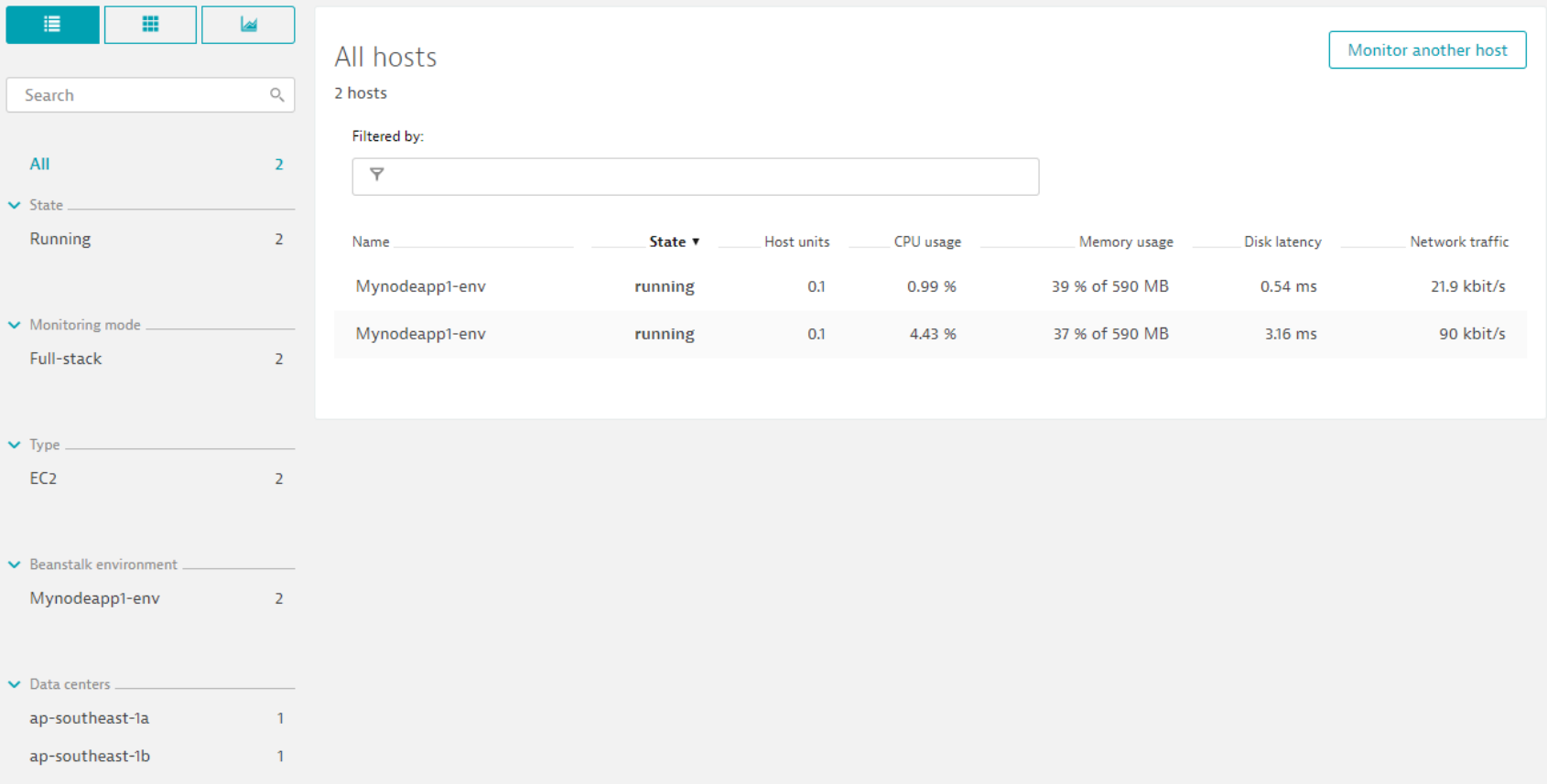
Change

Recent Events

Show All

Time	Type	Details
2017-10-24 20:05:32 UTC+0800	INFO	Environment health has transitioned from Info to Ok.
2017-10-24 20:02:32 UTC+0800	INFO	Environment health has transitioned from Ok to Info. Command is executing on 1 out of 2 instances.
2017-10-24 20:02:16 UTC+0800	INFO	Environment update completed successfully.
2017-10-24 20:02:16 UTC+0800	INFO	Successfully deployed new configuration to environment.
2017-10-24 20:01:32 UTC+0800	INFO	Added instance [ ] to your environment.





Explore User Sessions!

Dashboards & reports
- Dashboards
- Create custom chart
- Reports

Analyze
- Problems
- User sessions
- Log files
- Smartscape topology
- Diagnostic tools

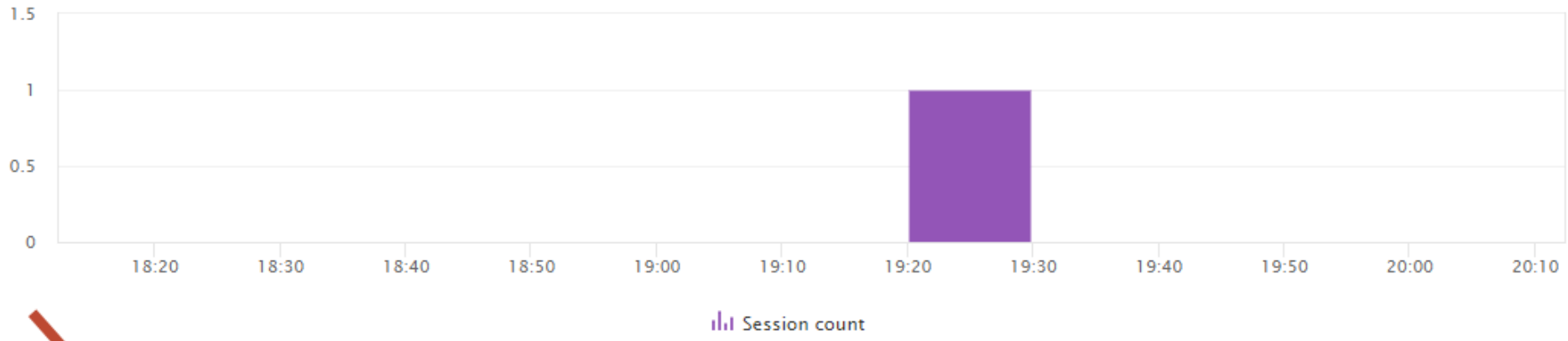
Monitor
- Applications
- Synthetic availability
- Transactions & services
- Databases
- Hosts
- Network
- Technologies

User sessions

Start typing to select user-session attributes and values for analysis...

User sessions

1 user session matching the search criteria. Chart results over time . Show a resolution of 5 minutes

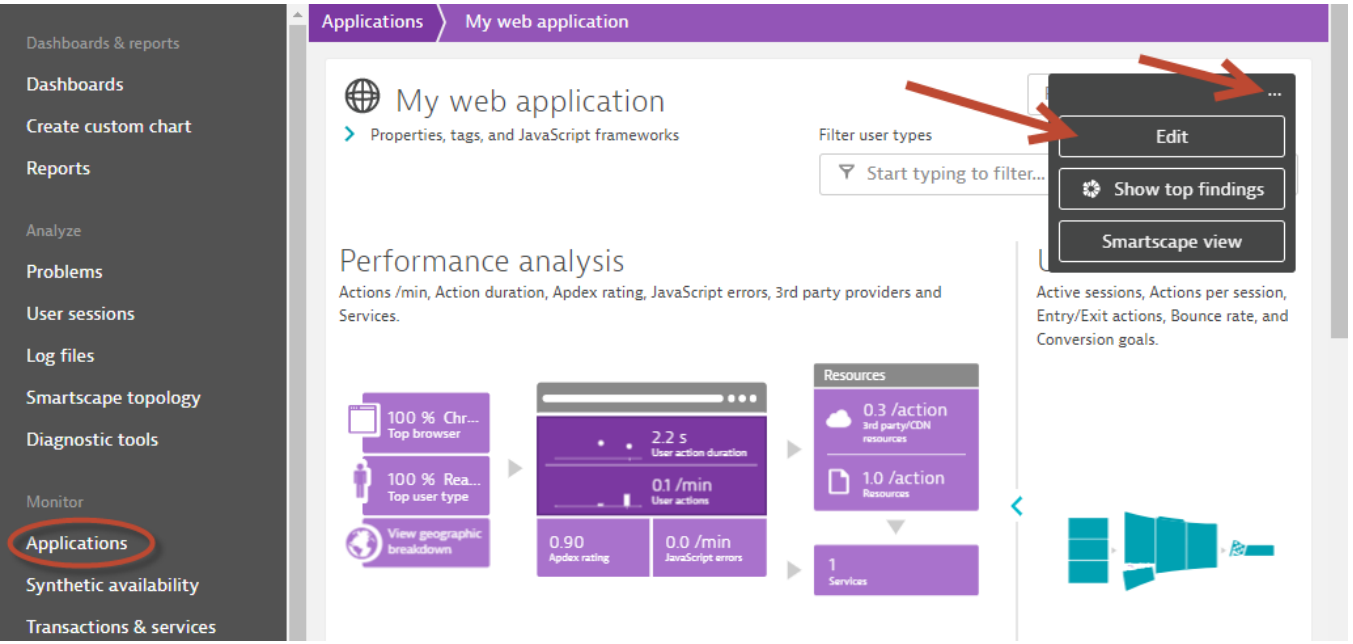


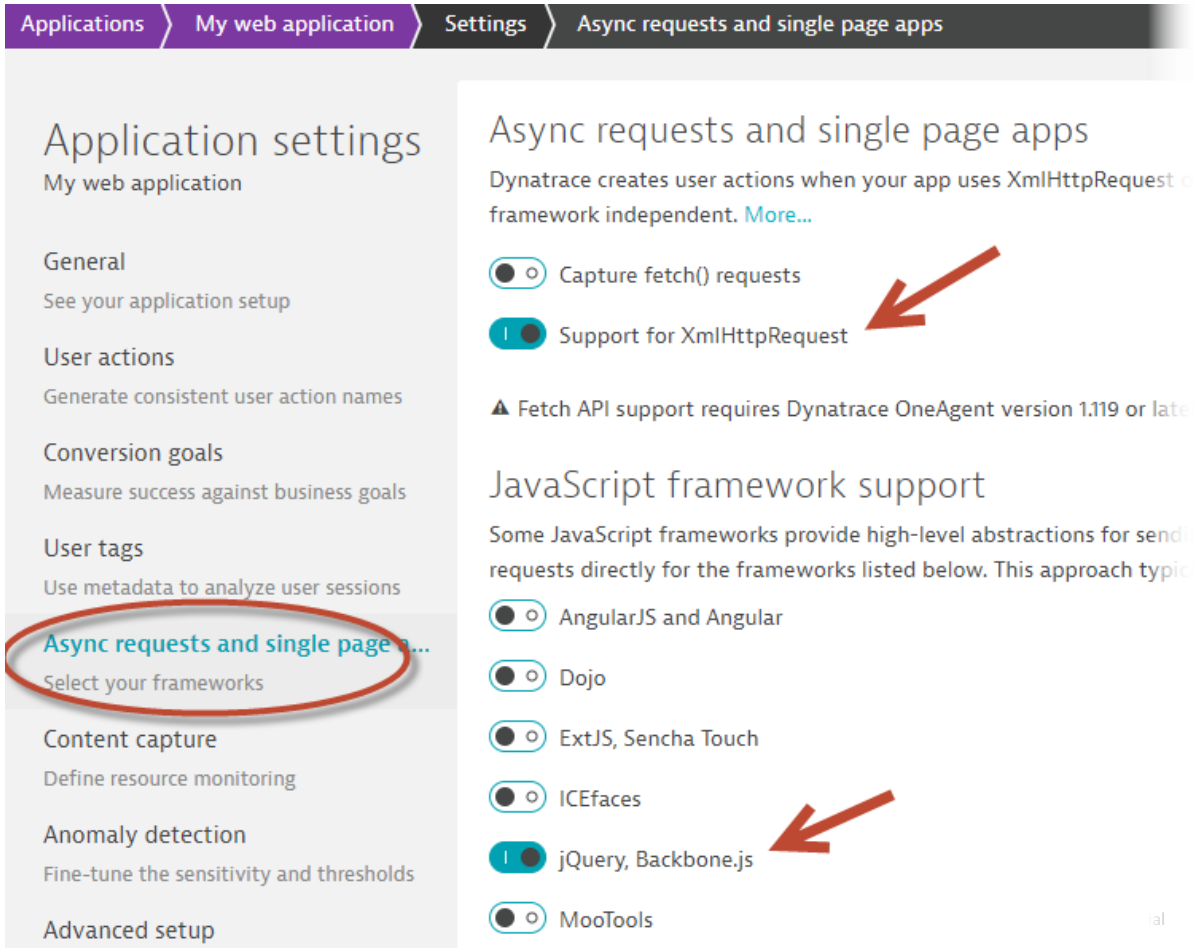
Showing last 1 of 1 users with 1 user sessions

Type	User	Location	Avg. actions / session	Avg. session duration	Session count ▾	Errors & crashes
	Anonymous (1508844486776CJ547MRUL7D7H323ALET9HCCAJ3B31RF)	Singapore, Central Singapore Community Development Council, Singapore	3.00	12 s	1	0

Additional Step: Real User Monitoring

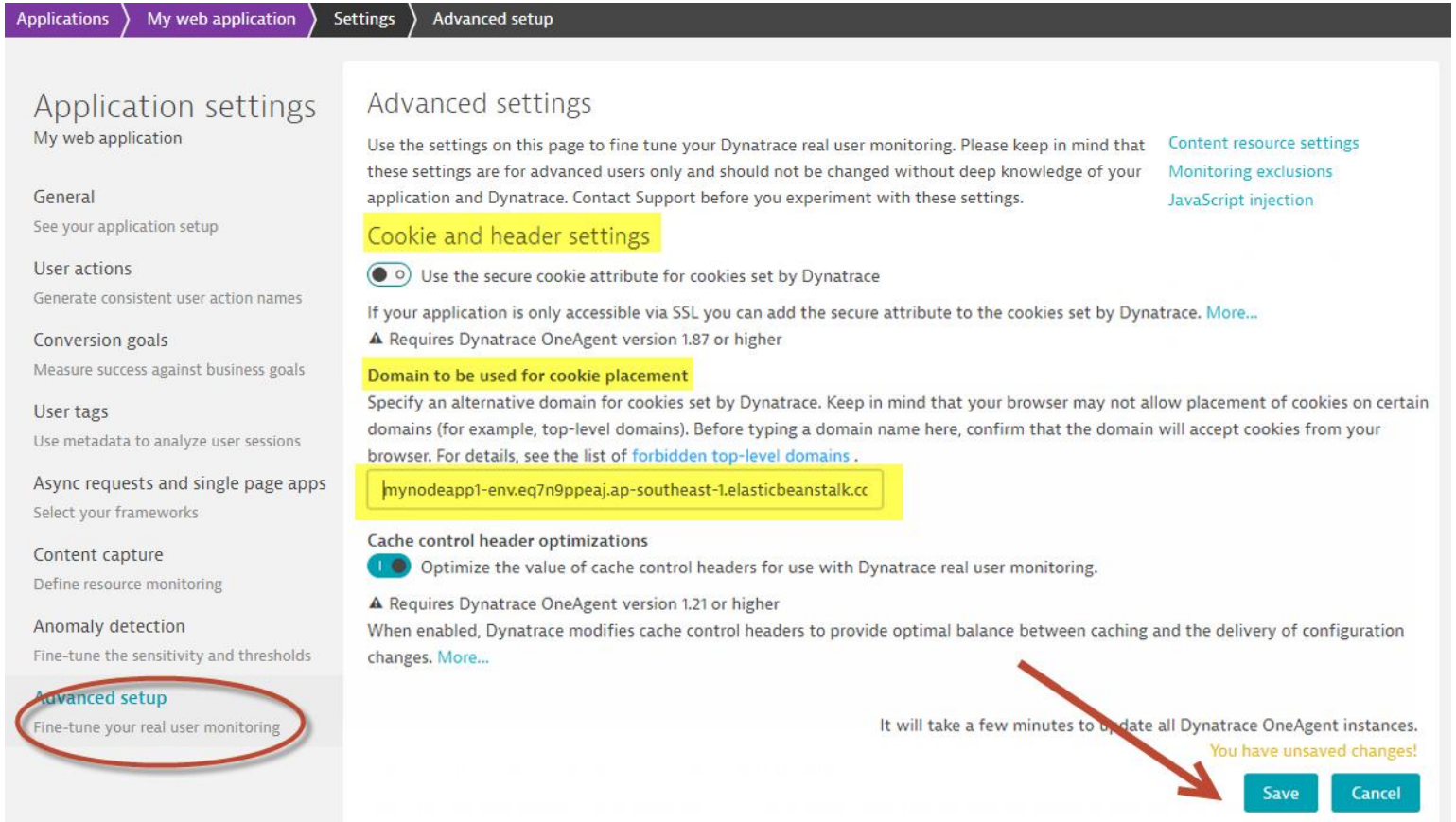
- Enable jQuery support for our Beanstalk Application
- My web application >> Settings >> Async requests and single page apps





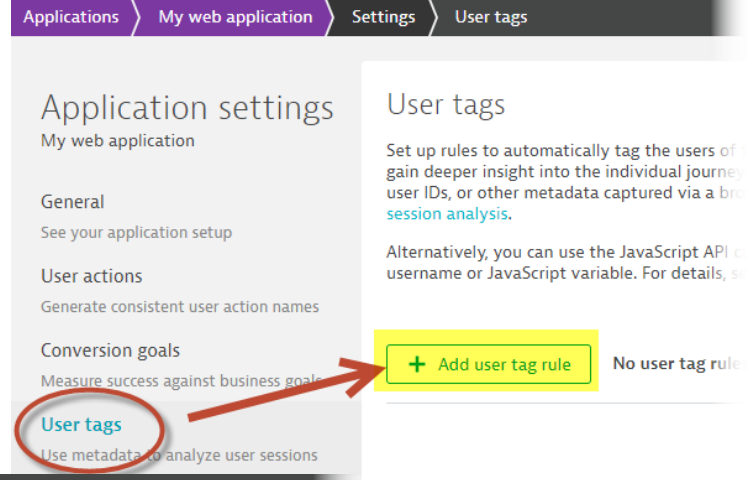
Additional Step: Real User Monitoring

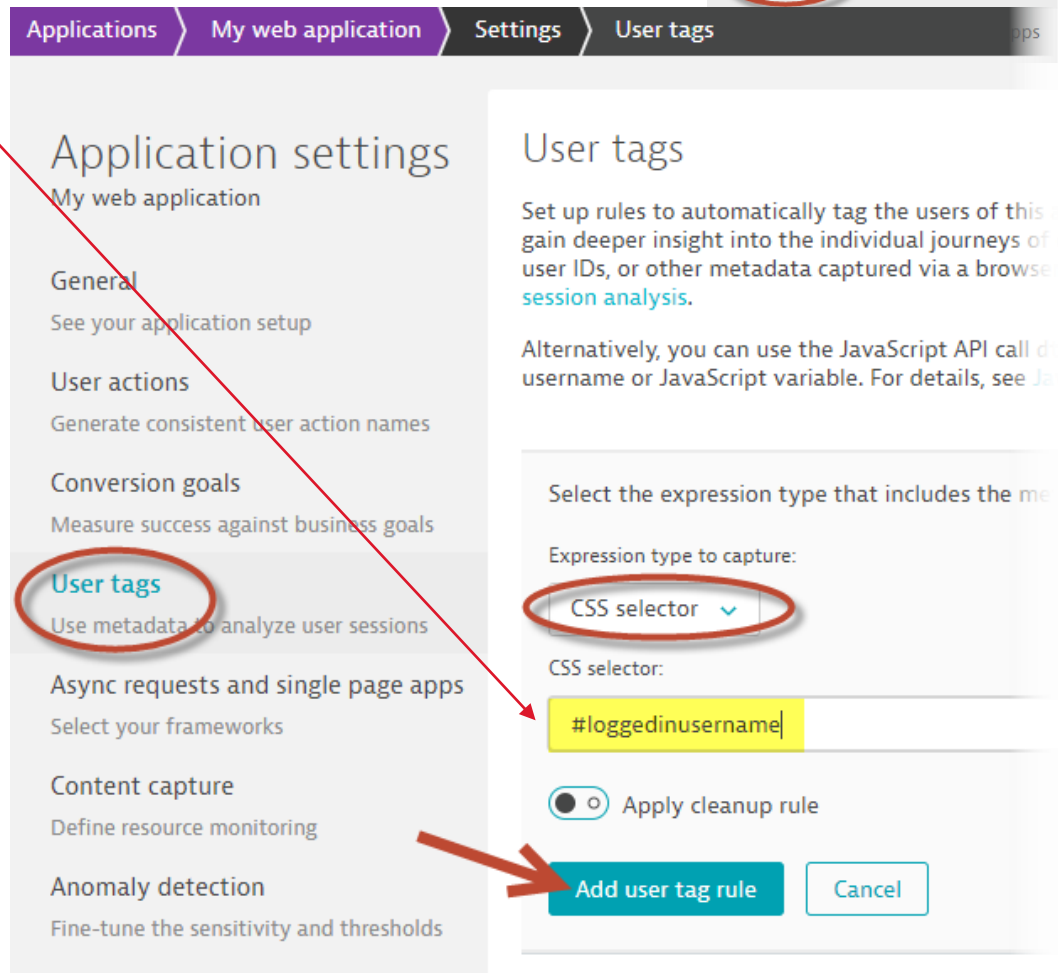
- Configure Cookie Support for elasticbeanstalk.com domains
  - Advanced Settings
  - Specify your full domain name, e.g: custom-env.ub2cp9hmpy.us-west-2.elasticbeanstalk.com in the field Domain to be used for cookie placement
- Why? Browser will reject the Dynatrace Cookies necessary for end user tagging
- This step IS NOT necessary if you host your app on a "normal" domain!



Additional Step: Real User Monitoring

- Configure User Tagging
  - Application has a login button which will then set the Username to an HTML Element with the ID #loggedinusername
  - Use CSS Selectors
- NOTE: User Visits right now will show up once the Visits are completed which means after the 30 minutes timeout!





User sessions

Start typing to select user-session attributes and values for analysis...

User sessions

2 user sessions matching the search criteria. Chart results over time . Show a resolution of 5 minutes

Showing last 2 of 2 users with 2 user sessions

Type	User	Location	Avg. actions / session	Avg. session duration	Session count ▾	Errors & crashes
	Anonymous (1508844486776CJ547MRUL7D7H323ALET9HCCAJ3B31RF)	Singapore, Central Singapore Community Development Council, Singapore	3.00	12 s	1	0
	Joe Mueller	Singapore, Central Singapore Community Development Council, Singapore	7.00	1 min 50 s	1	0

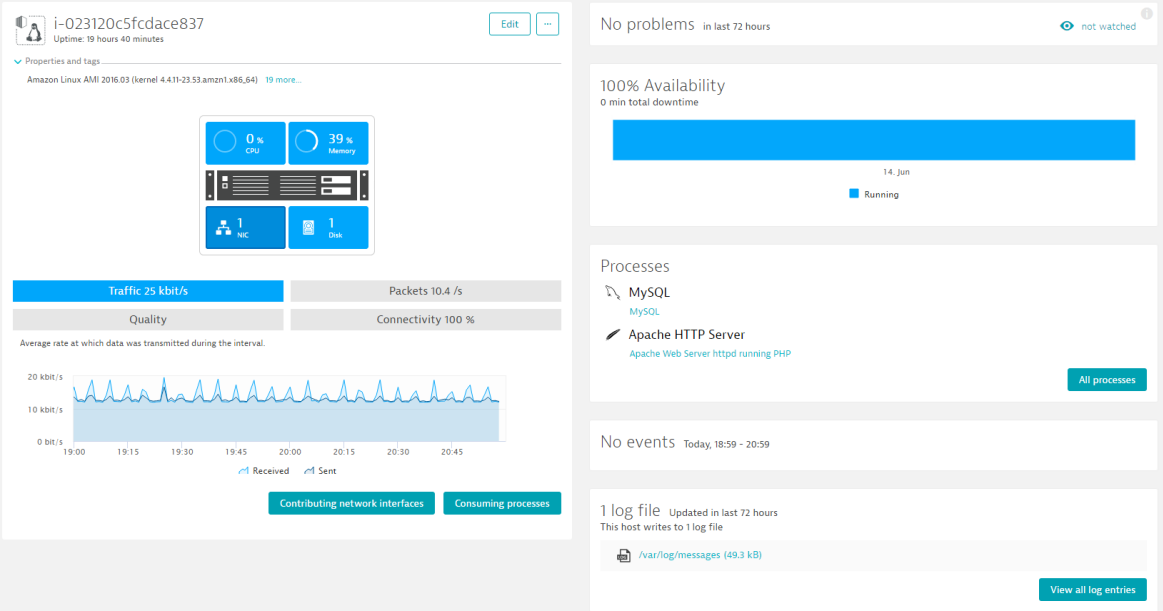
AWS Hands On 3

Goals: Deploying EC2 instance with a OneAgent

Goal: See EC2 Host monitored by Dynatrace

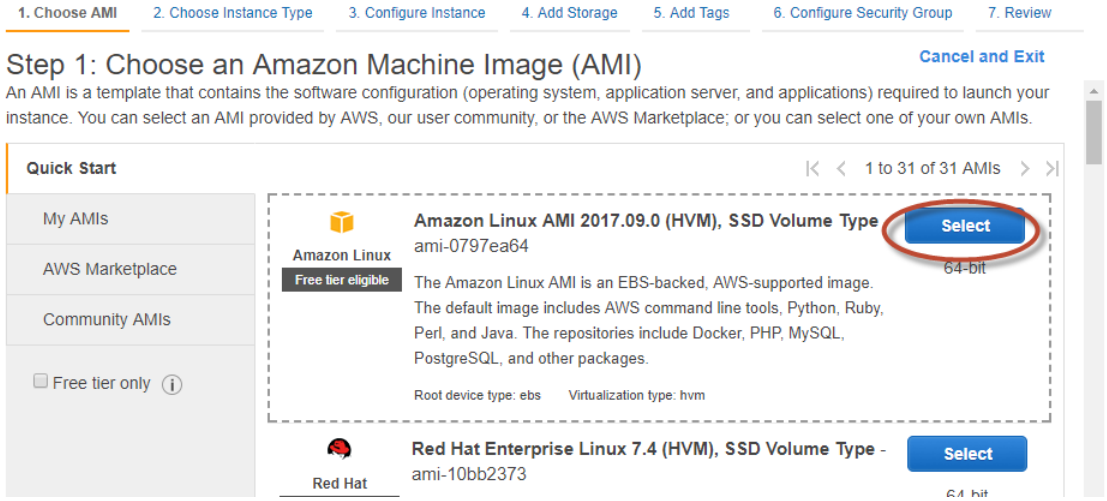
- 2 different options available to deploy a Dynatrace OneAgent on EC2
  - Through Puppet, Chef, Ansible, AWS CodeDeploy
  - Through UserData which is a “Startup Script” executed when launching EC2 Instance
- We will be using UserData Option

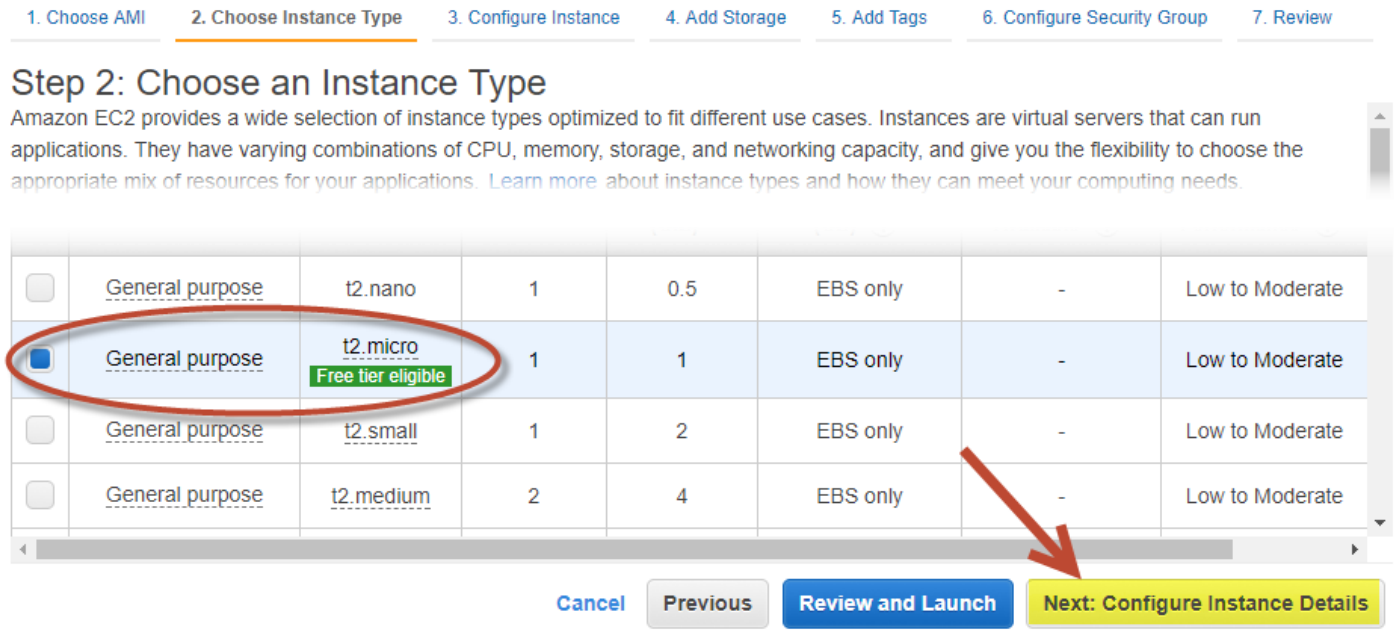
```
#!/bin/bash
 wget -O Dynatrace-OneAgent-Linux.sh https://YOUR.FULL.DYNATRACE.ONEAGENT.DOWNLOADLINK
/bin/sh Dynatrace-OneAgent-Linux.sh APP_LOG_CONTENT_ACCESS=1 INFRA_ONLY=0
```



Step 1

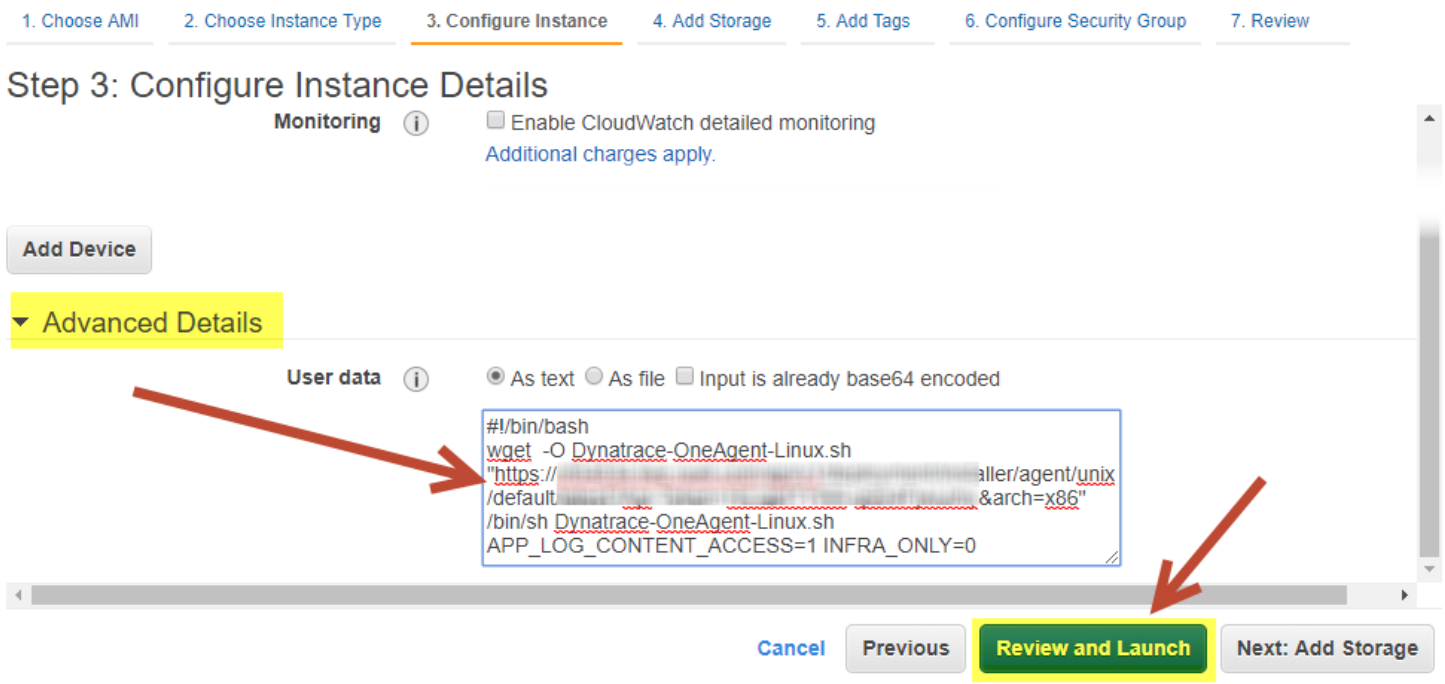
- Logon to AWS and navigate to EC2
- Select the option to Launch a new Instance
- Select Amazon Linux AMI and then select the free tier eligible t2.micro instance type





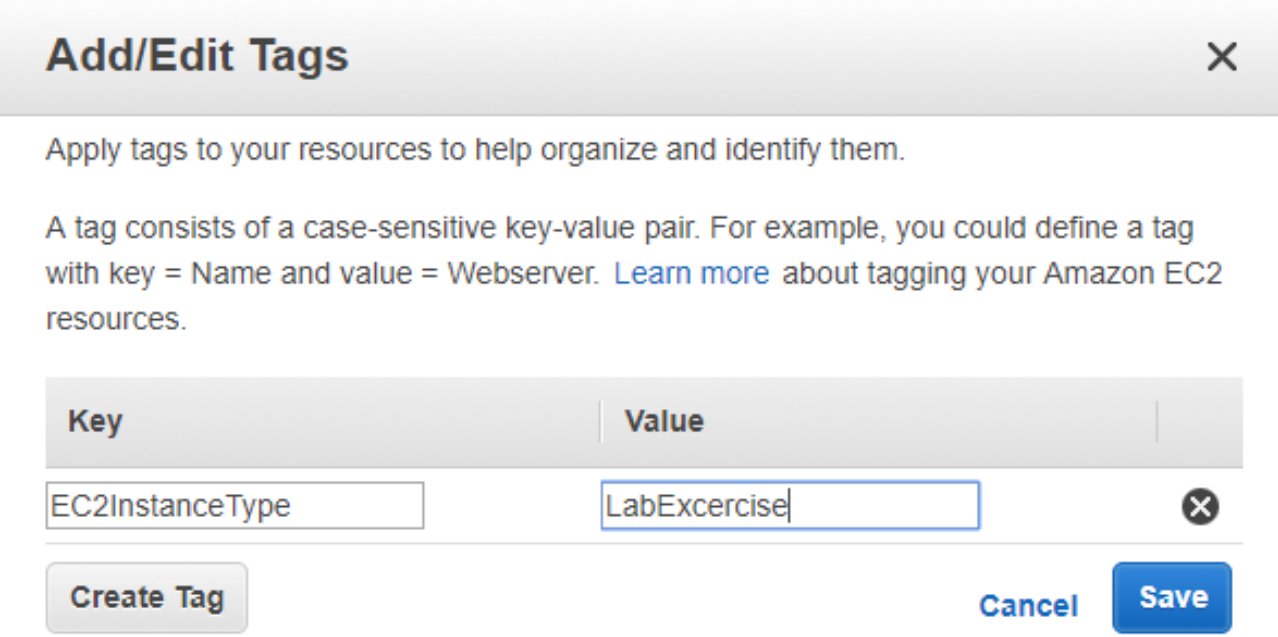
- Select Next Configure Instance Details
- Configure Instance: Expand the Advanced section and specify the following User Data script (make sure you use your unique OneAgent Download URI)

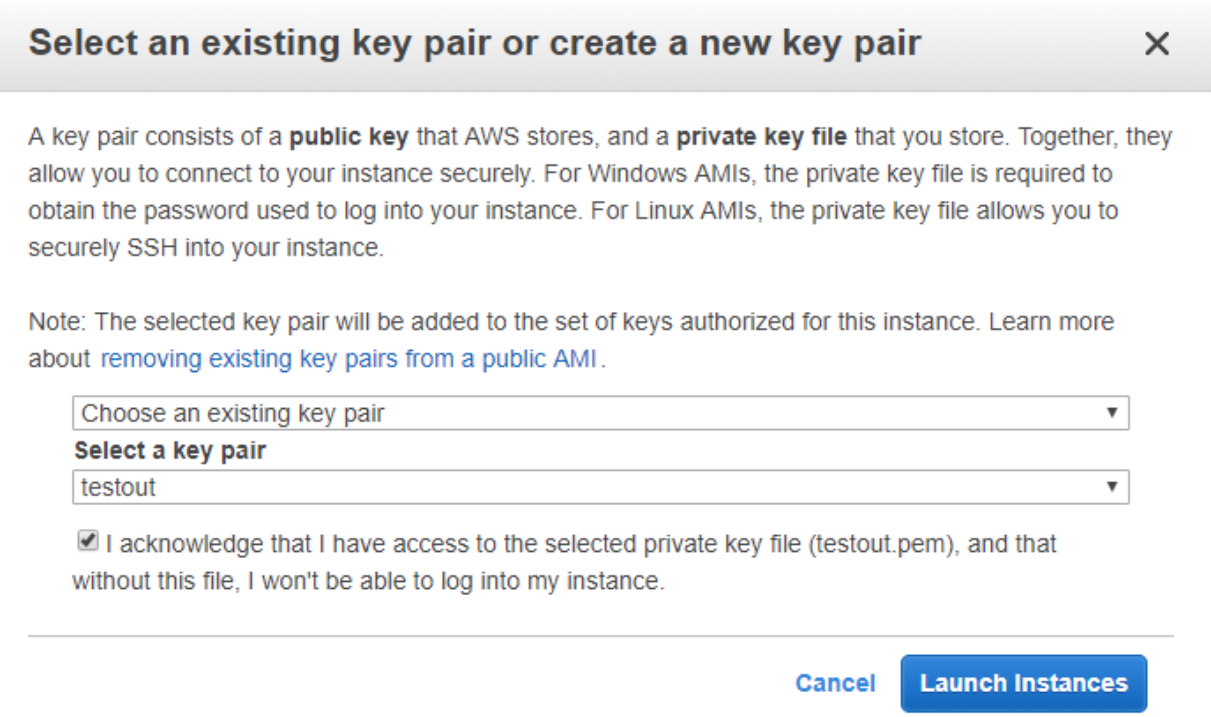
```
#!/bin/bash
 wget -O Dynatrace-OneAgent-Linux.sh https://YOUR.FULL.DYNATRACE.ONEAGENT.DOWNLOADLINK
/bin/sh Dynatrace-OneAgent-Linux.sh APP_LOG_CONTENT_ACCESS=1 INFRA_ONLY=0
```



Step 2

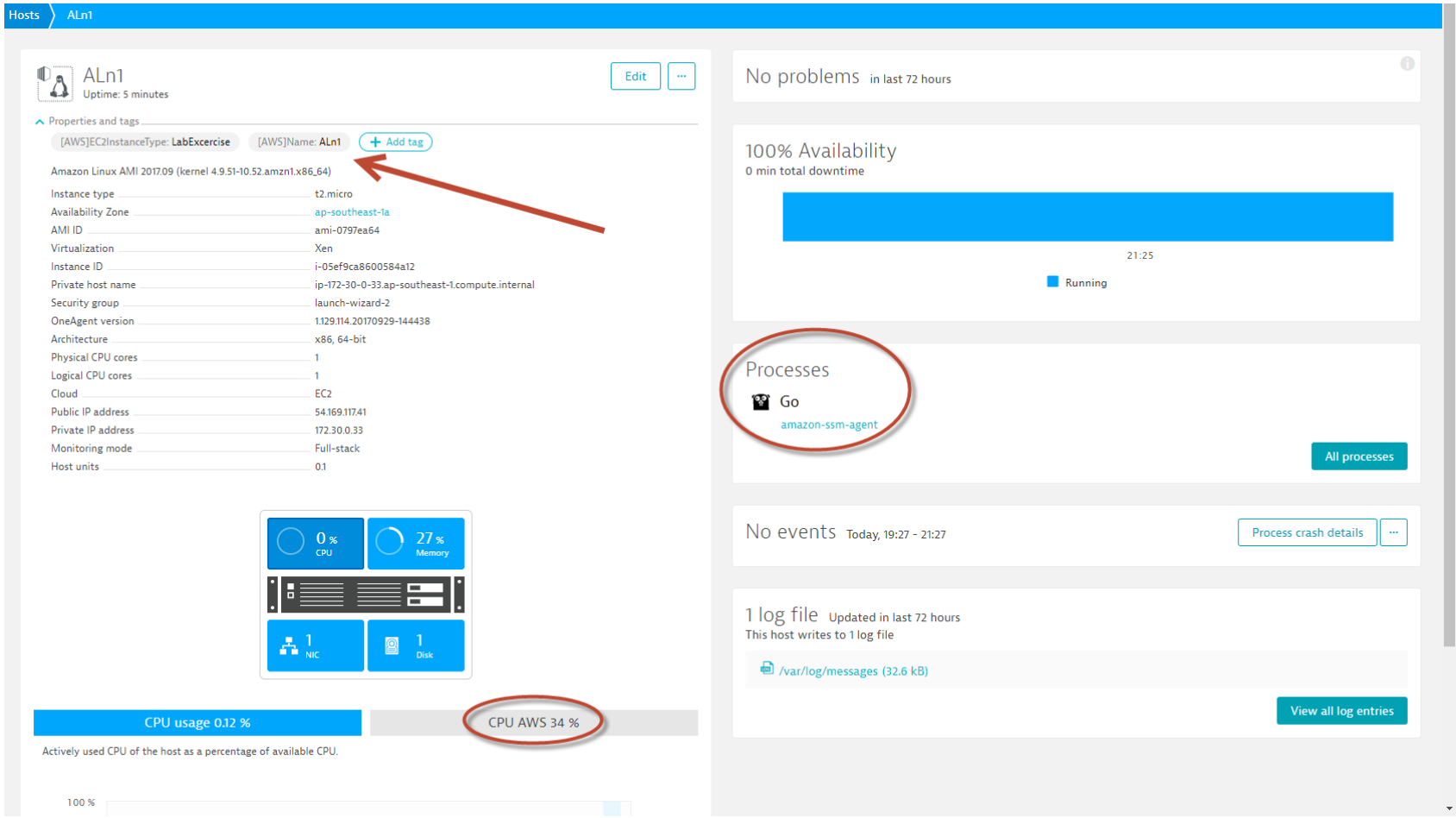
- Add Tags: on this configuration screen we add a custom tag. Key=EC2InstanceType; Value=LabExcercise.
- Click through the rest of the steps. Review settings and click Launch
- Select or create a new key pair. We will need this for remoting into EC2
- You can observe the launch log





Explore Dynatrace!

- Navigate to the Dynatrace Hosts list and wait until the host shows up. Click on it and explore what is monitored
- Expand the list of Properties and Tags. We should also find our EC2InstanceType tag with the value LabExcercise



Resources

- Dynatrace online help: https://help.dynatrace.com
  - Specifically for AWS: https://help.dynatrace.com/infrastructure/amazon-web-services/how-do-i-start-amazon-web-services-monitoring
- GitHub online tutorial for AWS: https://github.com/Dynatrace/AWSMonitoringTutorials
- Dynatrace blog: https://www.dynatrace.com/blog
- PurePerformance podcasts: https://www.dynatrace.com/community/pureperformance
- Dynatrace webinars: https://www.dynatrace.com/company/webinars


dynatrace
Every user, every app, everywhere.