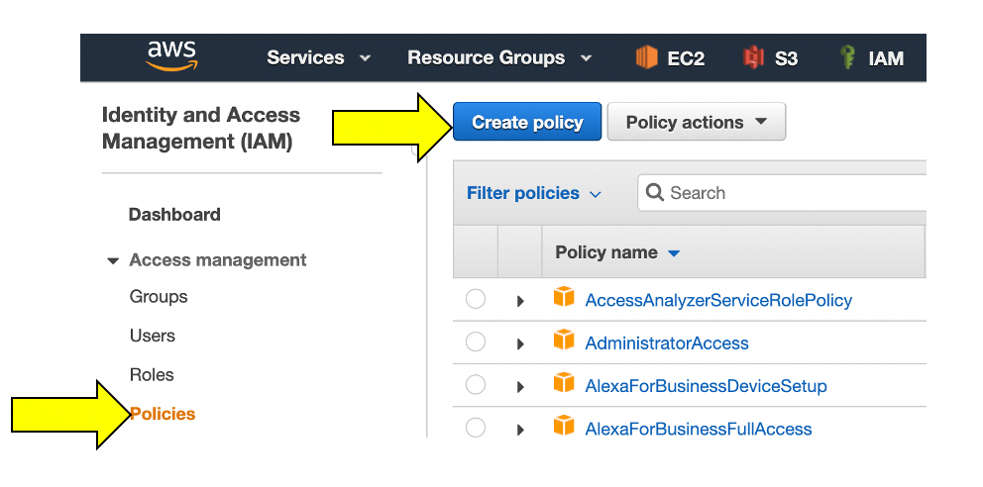
Step 1 of 3: Create AWS monitoring policy

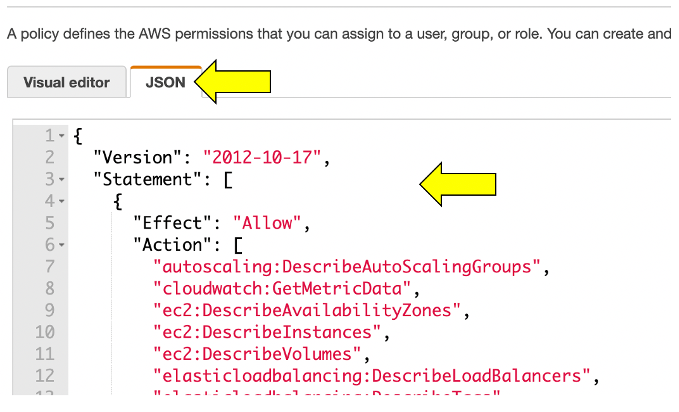
The AWS monitoring policy defines the minimum scope of permissions you need to give to Dynatrace to monitor the services running in your AWS account. Create it once and use anytime when enabling Dynatrace access to your AWS account.

1 . Go to **Identity and Access Management (IAM)** in your Amazon Console.

2 . Go to **Policies** and click the Create policy button.

[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-policy.png)

3 . Select the JSON tab, and paste the predefined policy from the box below.

[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-policy-json.png)

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"autoscaling:DescribeAutoScalingGroups",

"cloudwatch:GetMetricData",

"ec2:DescribeAvailabilityZones",

"ec2:DescribeInstances",

"ec2:DescribeVolumes",

"elasticloadbalancing:DescribeLoadBalancers",

"elasticloadbalancing:DescribeTags",

"elasticloadbalancing:DescribeInstanceHealth",

"elasticloadbalancing:DescribeListeners",

"elasticloadbalancing:DescribeRules",

"elasticloadbalancing:DescribeTargetHealth",

"rds:DescribeDBInstances",

"rds:DescribeEvents",

"rds:ListTagsForResource",

"dynamodb:ListTables",

"dynamodb:ListTagsOfResource",

"lambda:ListFunctions",

"lambda:ListTags",

"elasticbeanstalk:DescribeEnvironments",

"elasticbeanstalk:DescribeEnvironmentResources",

"s3:ListAllMyBuckets",

"sts:GetCallerIdentity",

"cloudformation:ListStackResources",

"tag:GetResources",

"tag:GetTagKeys",

"cloudwatch:ListMetrics",

"kinesisvideo:ListStreams",

"sns:ListTopics",

"sqs:ListQueues",

"ec2:DescribeNatGateways",

"ec2:DescribeSpotFleetRequests",

"kinesis:ListStreams",

"es:ListDomainNames",

"cloudfront:ListDistributions",

"firehose:ListDeliveryStreams",

"elasticmapreduce:ListClusters",

"kinesisanalytics:ListApplications",

"elasticache:DescribeCacheClusters",

"elasticfilesystem:DescribeFileSystems",

"ecs:ListClusters",

"redshift:DescribeClusters",

"rds:DescribeDBClusters",

"glue:GetJobs",

"sagemaker:ListEndpoints",

"apigateway:GET"

],

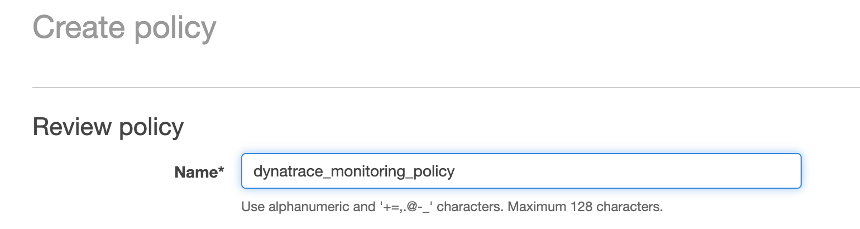
"Resource": "\*"

}

]

}

4 . Use the policy name of dynatrace\_monitoring\_policy

[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-policy-name.png)

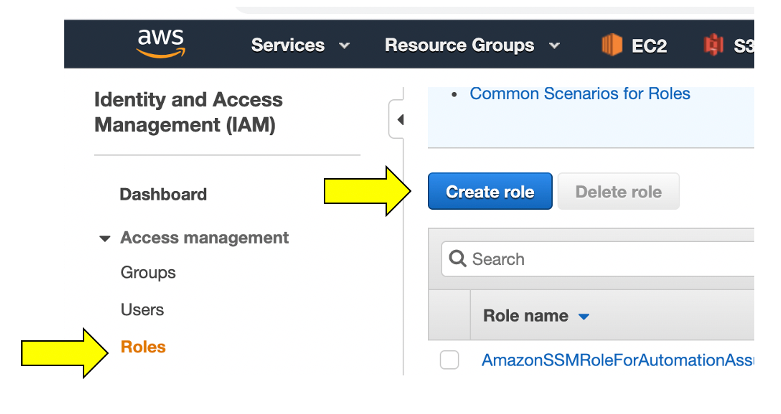
5 . Click **Create policy** button.

Step 2 of 3: Create an AWS monitoring role and attach monitoring policy

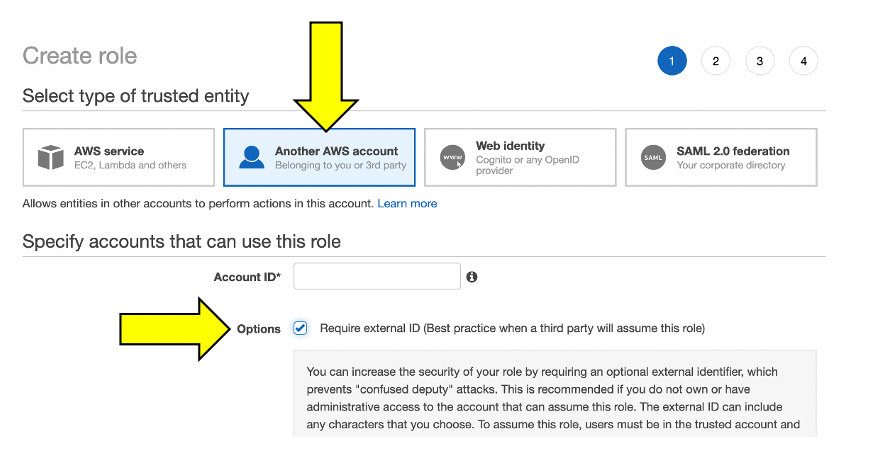
To give Dynatrace SaaS the role-based monitoring access to your AWS account, you need to create a dedicated monitoring role for Dynatrace in your AWS account. Dynatrace will use this role to authenticate in your AWS environment with the scope of permissions as defined by the monitoring policy.

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

2 . Go to Roles and click the **create role** button.

[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-role.png)

3 . Select the **Another AWS account** tile as to establish trust with the Dynatrace account. Also select the **Require external ID** option button

[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-role-trust-account.png)

The AWS Account ID for Dynatrace SaaS (e.g. [https://YOUR\_TENANT.live.dynatrace.com/](https://your_tenant.live.dynatrace.com/)) is **509560245411**. This AWS Account ID is the account that the role within the AWS account you are using for the workshop will use. The Token generated on the Dynatrace AWS connection page, that is used in the external ID field for the AWS role, adds another level of security so that this role can only access the data for your specific Dynatrace tenant and Dynatrace connection. If you were using your own Dynatrace managed cluster, this AWS Account ID value would be different.

To get the Account and External ID for the AWS role, in your browser, open a new tab and sign in to Dynatrace.

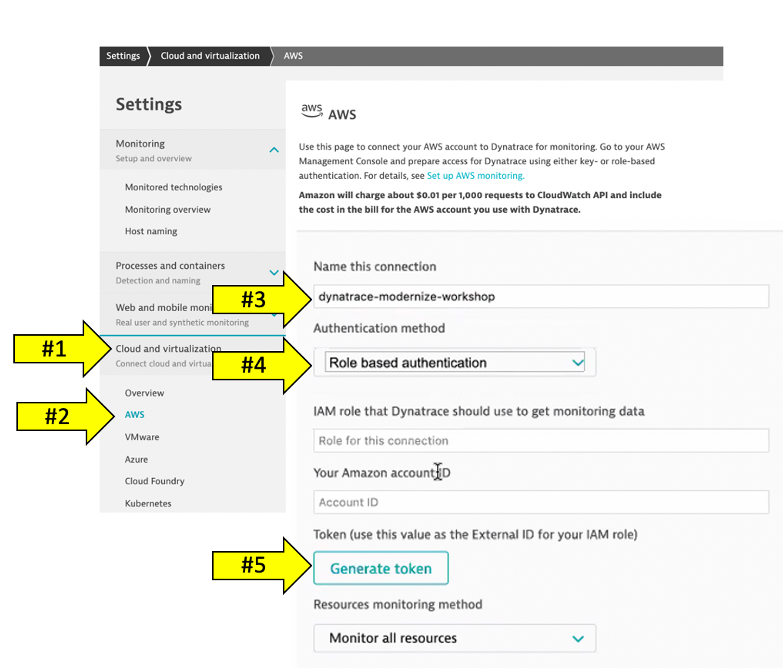
1 . Once logged in, go to **Settings > Cloud and virtualization**

2 . Choose **AWS** menu then click **Connect new instance** button

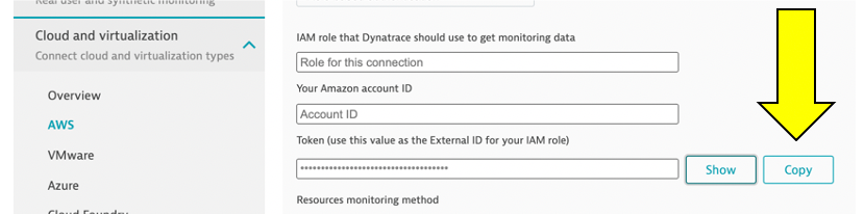
3 . Enter the name for this connection as dynatrace-modernize-workshop

4 . select **Role based authentication** method.

5 . click the **Generate token** button next.

[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-dt.png)

6 . Leave **IAM role** and **AWS Account** blank for now and click the **Copy** button next to the generated token.

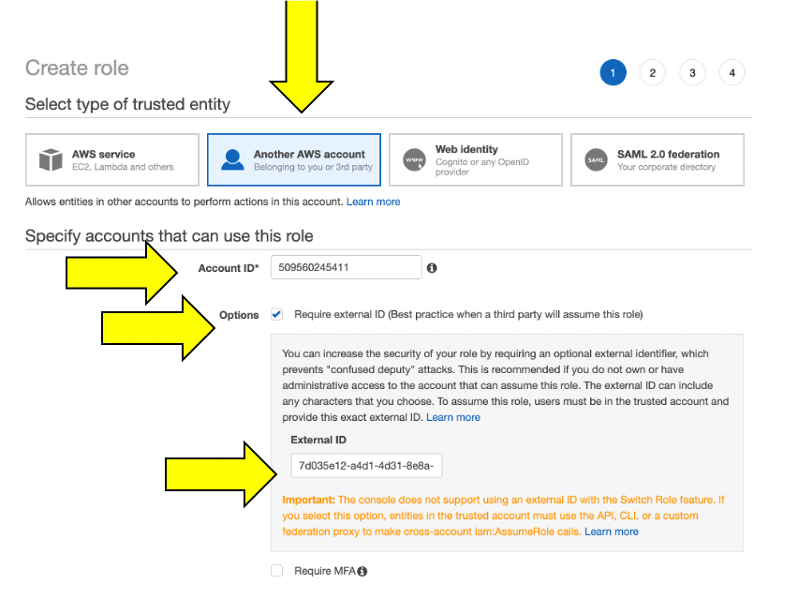
[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-dt-copy.png)

Keep this browser window open. We will come back to it shortly to copy the Token and test the connection.

Now back in the AWS console tab **Create Role** page, enter these values:

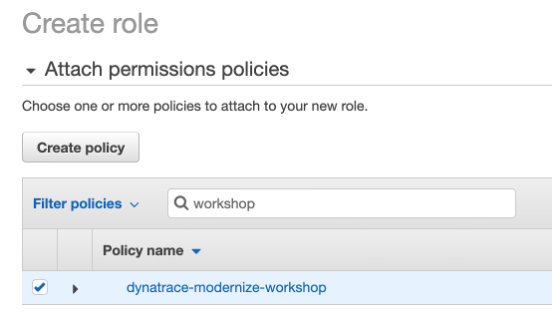
* Paste the **Token** value that you just copied from your Dynatrace AWS connection page to the **External ID** field
* Copy 509560245411 this number to the Account ID field

Your **Select type of trusted entity** page should look like this:

[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-role-account.png)

7 . Now click the **Next: Permissions** button on the bottom

8 . On the **Attach permissions policies** page, choose the monitoring policy you created, search for: dynatrace\_monitoring\_policy. Choose then checkbox next to it and then click the Next: Review button.

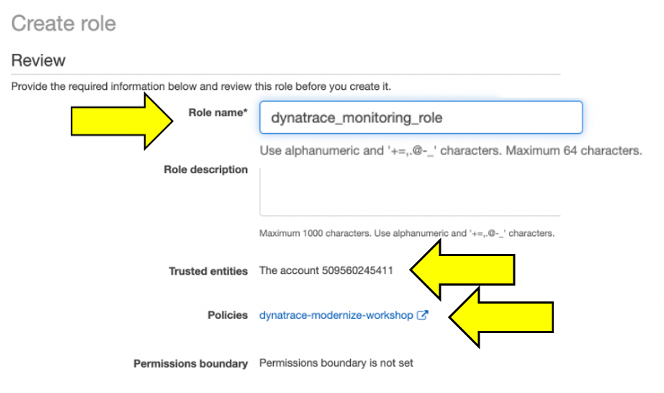
[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-role-policy.png)

9 . Now click the **Next: Tags** button on the bottom

10 . On the **Add Tags** page, leave the defaults and click the **Next: Review** button on the bottom

11 . On the **Review** page, provide the role name of: dynatrace\_monitoring\_role.

The role name, trusted entities and policy should have values as shown here.

[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-role-policy-name.png)

9 . Click the **Create Role** button.

Step 3 of 3: Connect

Now that we have the AWS role, it’s time to connect Dynatrace to your Amazon AWS account using that role.

1 . Go back on the Dynatrace connection page you still have open

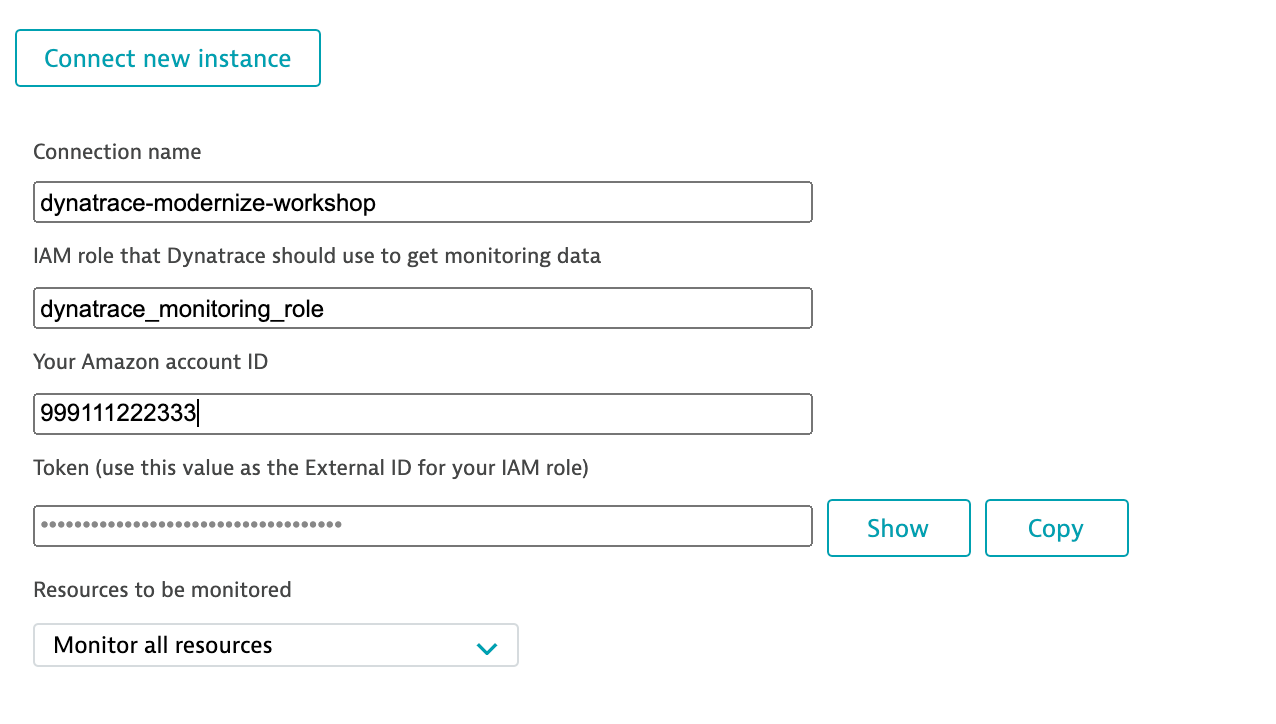
2 . In the Role field, use the name of the role you created easier: dynatrace\_monitoring\_role

3 . Type your Account ID (the account you want us to pull metrics from). To get your Account ID, go to your Cloud9 IDE and type in this command and copy the **Account** value from the JSON output as shown here.

aws sts get-caller-identity

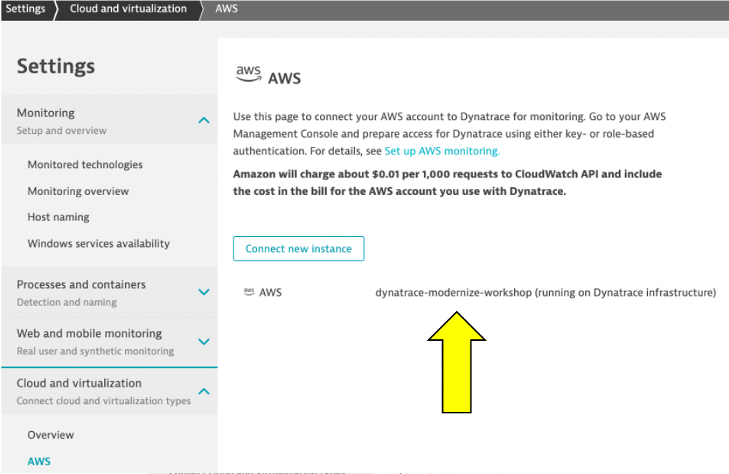
[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-account.png)

4 . Your Connection page should now look like this:

[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-final.png)

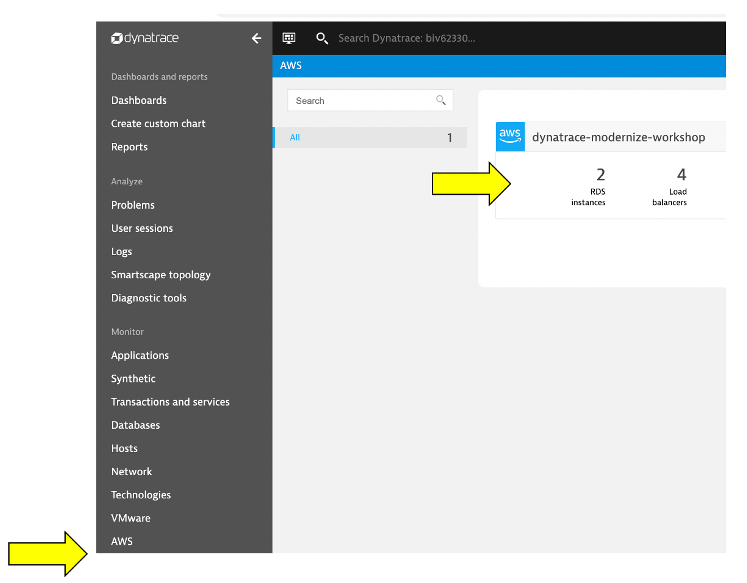
5 . Click the **Connect** button to verify and once verified, click the **save** button.

If successful, your should see the configuration now on the AWS connections page:

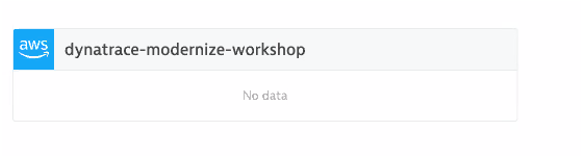
[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-list.png)

Review the AWS monitor page

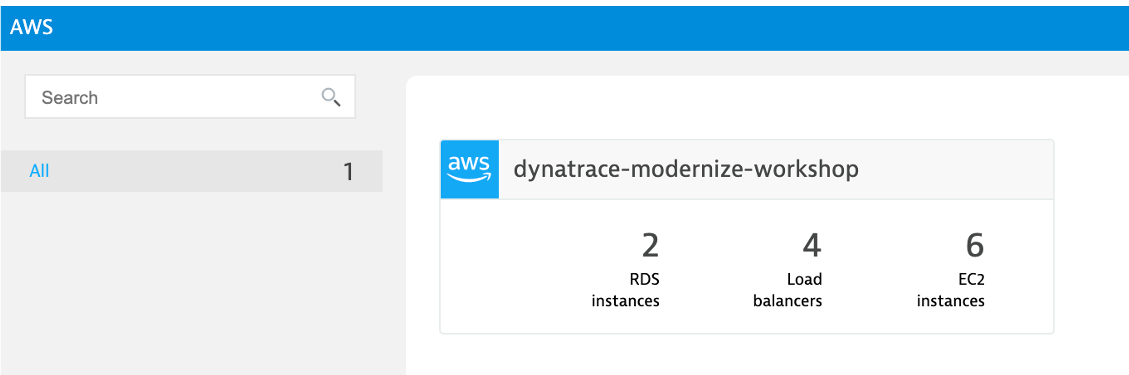
On the far left Dynatrace menu, navigate to the “AWS” menu.

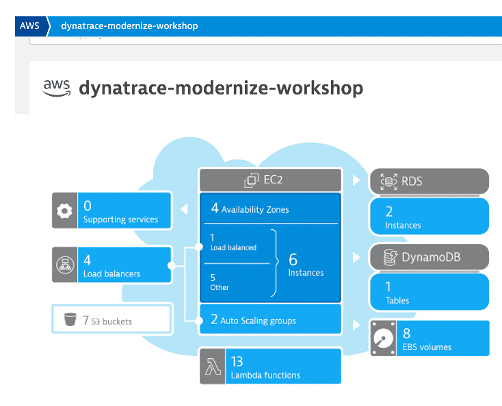
[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-menu.png)

You may see “no data” initially as seen here. This is because Dynatrace makes Amazon API requests every 5 minutes, so it might take a few minutes for data to show untill we are done with application setup on AWS.

[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-blank.png)

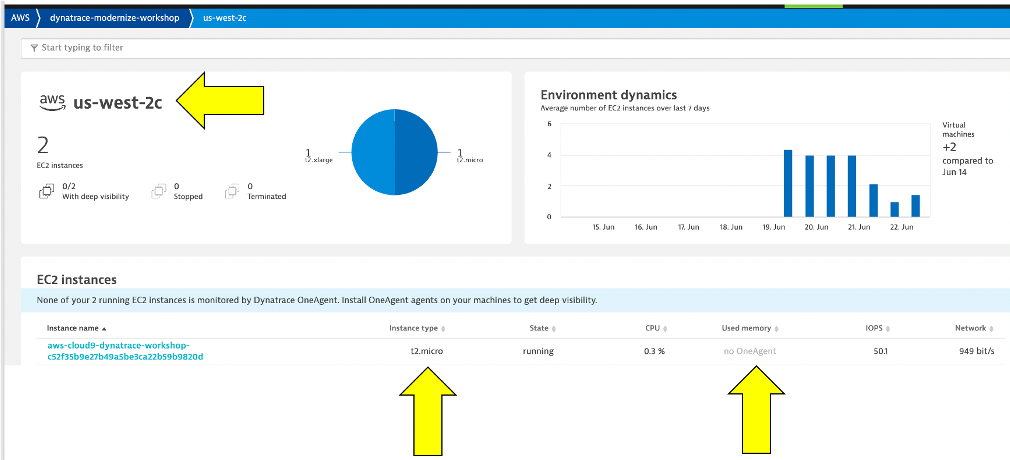
Once data is coming in, the dashboard pages will look similar to what is shown below.

[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard-overview.png)

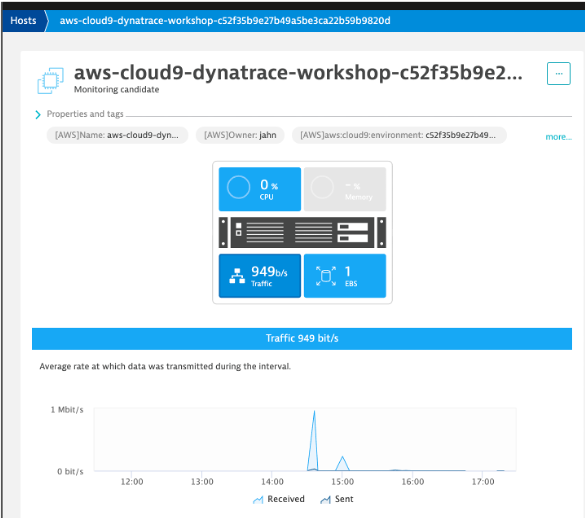
[](https://dynatrace.awsworkshop.io/images/dt-aws-dashboard.png)

Review your metrics

Once data starts to be collected, click in the blue availability zone section located under the grey header labeled EC2 and you should see the list of availability zones below. Click on **us-west-2c** and the EC2 instances will be listed. You should find the Cloud9 instance which does NOT have a Dynatrace OneAgent running on it. Notice too how you automatically get regional and instance type data.

[](https://dynatrace.awsworkshop.io/images/aws-monitor-list.png)

Click on the **Cloud9** instance, and you will see how this host still is represented in the same **Host** view that we saw earlier with the host running the OneAgent. The basic CPU and memory metrics from CloudWatch are graphed for you. What is GREAT, is that this host is being monitored automatically by the Dynatrace AI engine and can raise a problem when there are anomalies.

[](https://dynatrace.awsworkshop.io/images/aws-monitor-host.png)