

100 Days of DevOps — Day 18-Add monitoring to these instances using Terraform(CloudWatch and SNS)

Check the updated 101 Days of DevOps Course

Course Registration link: <https://www.101daysofdevops.com/register/>

Course Link: <https://www.101daysofdevops.com/courses/101-days-of-devops/>

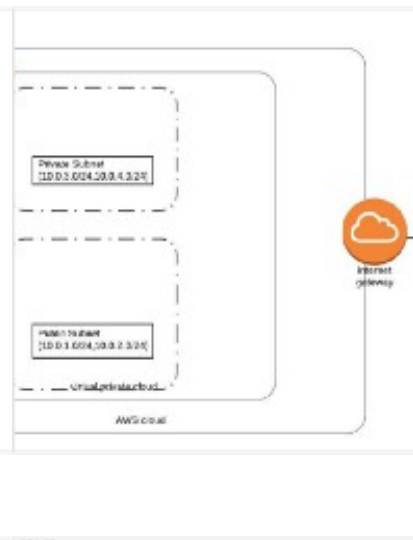
YouTube link: <https://www.youtube.com/user/laprashant/videos>

Welcome to Day 18 of 100 Days of DevOps, Let continue our journey, so far we have discussed fundamentals of terraform, build VPC and EC2 instance using terraform, today let's add monitoring piece to it

100 Days of DevOps — Day 15- Introduction to Terraform

Welcome to Day 15 of 100 Days of DevOps, Let continue our journey and focus on Automation especially on Infrastructure...


medium.com



100 Days of DevOps — Day 16- Building VPC using Terraform

Welcome to Day 16 of 100 Days of DevOps, Let continue our journey, yesterday I discussed terraform, today let's build...

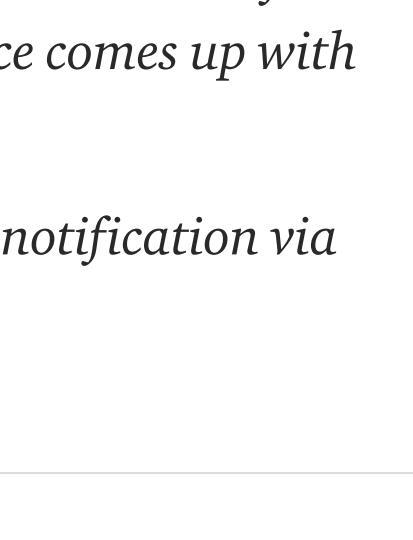
medium.com



100 Days of DevOps — Day 17- Creating EC2 Instance using Terraform

Welcome to Day 17 of 100 Days of DevOps, Let continue our journey, so far I havr discussed terraform fundamentals and...

medium.com



- Adding monitoring piece can be achieved via two ways, we can add separate cloudwatch and SNS module and then call it in our EC2 module
- We can call the Cloudwatch and SNS terraform code directly in EC2 terraform module, I prefer this approach as we want all our EC2 instance comes up with monitoring enabled
- First, let start with SNS topic, which is required to send out a notification via Email, SMS when an event occurs.

```
1 resource "aws_sns_topic" "alarm" {
2   name = "alarms-topic"
3
4   delivery_policy = <<EOF
5 {
6   "http": {
7     "defaultHealthyRetryPolicy": {
8       "minDelayTarget": 20,
9       "maxDelayTarget": 20,
10      "numRetries": 3,
11      "numMaxDelayRetries": 0,
12      "numMinDelayRetries": 0,
13      "backoffFunction": "linear"
14    },
15    "disableSubscriptionOverrides": false,
16    "defaultThrottlePolicy": {
17      "maxReceivesPerSecond": 1
18    }
19  }
20 }
21 }
22 EOF
23
24 provisioner "local-exec" {
25   command = "aws sns subscribe --topic-arn ${self.arn} --protocol email --notification
26 }
27 }
```

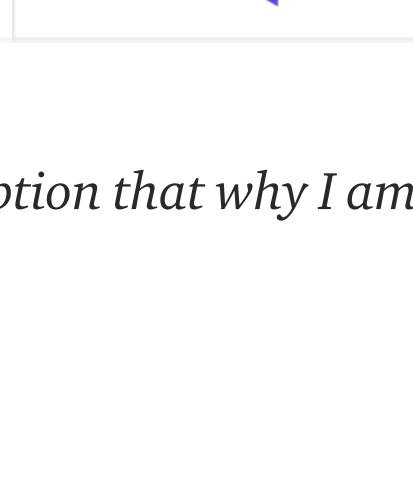
sns_topic.tf hosted with ❤️ by GitHub [view raw](#)

* Here I am trying to create an SNS topic resource
* give your SNS topic, some name
* After that I am using a default policy

AWS: sns_topic - Terraform by HashiCorp

Provides an SNS topic resource.

www.terraform.io



NOTE: As with SNS, someone needs to confirm the email subscription that why I am using local-exec provisioners with terraform.

Now let's take a look at terraform code for CloudWatch.

This code is divided into two parts

- Setup CPU Usage Alarm using the Terraform

```
1 resource "aws_cloudwatch_metric_alarm" "cpu-utilization" {
2   alarm_name         = "high-cpu-utilization-alarm"
3   comparison_operator = "GreaterThanOrEqualToThreshold"
4   evaluation_periods  = "2"
5   metric_name         = "CPUUtilization"
6   namespace           = "AWS/EC2"
7   period              = "120"
8   statistic            = "Average"
9   threshold            = "80"
10  alarm_description   = "This metric monitors ec2 cpu utilization"
11  alarm_actions        = [ "${aws_sns_topic.alarm.arn}" ]
12  dimensions {
13    InstanceId = "${aws_instance.my_instance.*.id[count.index]}"
14  }
15 }
```

cloudwatch_cpu_usage.tf hosted with ❤️ by GitHub [view raw](#)

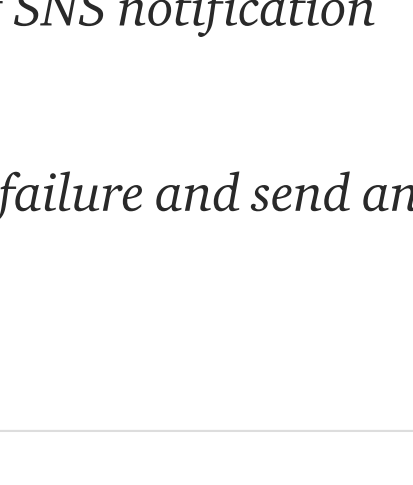
* Setup an alarm name
* This field is self explanatory,supported operators GreaterThanOrEqualToThreshold, GreaterThanThreshold, LessThanThreshold, LessThanOrEqualToThreshold.
* evaluation_period: The number of periods over which data is compared to the specified threshold(I setup 2 just for demo purpose but its completly depend upon your requirement)
* metric_name: Please check the link below list of services that publish Cloudwatch metrics
* namespace: The namespace for the alarm's associated metric(Check the second column of the link below)
* period: period in second(I am using 120 sec or 2min but again it completly depend upon your requirements)
* statistic: The statistic to apply to the alarm's associated metric, supported value: SampleCount, Average, Sum, Minimum, Maximum
* threshold: The value against which the specified statistic is compared.(I set it up as 80% i.e when CPU utilization goes above 80%)
* alarm_actions: The list of actions to execute when this alarm transitions into an ALARM state from any other state. Please note, each action is specified as an Amazon Resource Name (ARN)
* dimensions: The dimensions for the alarm's associated metric. Again check the below mentioned link for supported dimensions

AWS Services That Publish CloudWatch Metrics

AWS Services That Publish CloudWatch Metrics - Amazon CloudWatch

The following AWS services publish metrics to CloudWatch. For information about the metrics and dimensions, see the...

docs.aws.amazon.com



- So what this is doing this code is going to send an email using SNS notification when CPU Utilization is more than 80%.
- Now look at the other part is to perform system and instance failure and send an email using SNS notification

```
1 resource "aws_cloudwatch_metric_alarm" "instance-health-check" {
2   alarm_name         = "instance-health-check"
3   comparison_operator = "GreaterThanOrEqualToThreshold"
4   evaluation_periods  = "1"
5   metric_name         = "StatusCheckFailed"
6   namespace           = "AWS/EC2"
7   period              = "120"
8   statistic            = "Average"
9   threshold            = "1"
10  alarm_description   = "This metric monitors ec2 health status"
11  alarm_actions        = [ "${aws_sns_topic.alarm.arn}" ]
12  dimensions {
13    InstanceId = "${aws_instance.my_instance.*.id[count.index]}"
14  }
15 }
```

cloudwatch_instance_check.tf hosted with ❤️ by GitHub [view raw](#)

* Most of this code is almost similar, only difference is metric_name here is StatusCheckFailed

Final EC2 code with CloudWatch Monitoring and SNS topic enabled look like this

```
1 provider "aws" {
2   region = "us-west-2"
3 }
4
5 data "aws_availability_zones" "available" {}
6
7 # To get the latest Centos7 AMI
8 data "aws_ami" "centos" {
9   owners      = ["67959333241"]
10  most_recent = true
11
12  filter {
13    name   = "name"
14    values = ["CentOS Linux 7 x86_64 HVM EBS *"]
15  }
16
17  filter {
18    name   = "architecture"
19    values = ["x86_64"]
20  }
21
22  filter {
23    name   = "root-device-type"
24    values = ["ebs"]
25  }
26 }
27
28 resource "aws_key_pair" "mytest-key" {
29   key_name   = "my-test-terraform-key"
30   public_key = "${file(var.my_public_key)}"
31 }
32
33 resource "aws_instance" "test_instance" {
34   count          = "${var.instance_count}"
35   ami            = "${data.aws_ami.centos.id}"
36   instance_type  = "${var.instance_type}"
37   key_name       = "${aws_key_pair.mytest-key.id}"
38   vpc_security_group_ids = ["${var.security_group}"]
39   subnet_id      = "${element(var.subnet_id, count.index )}"
40   user_data      = "${data.template_file.user-init.rendered}"
41
42   tags {
43     Name = "my-test-server.${count.index + 1}"
44   }
45 }
46
47 resource "aws_ebs_volume" "my-test-ebs" {
48   count          = 2
49   availability_zone = "${data.aws_availability_zones.available.names[count.index]}"
50   size            = 10
51   type            = "gp2"
52 }
53
54 resource "aws_volume_attachment" "my-test-ebs-attachment" {
55   count          = 2
56   device_name    = "/dev/xvdb"
57   instance_id    = "${aws_instance.test_instance.*.id[count.index]}"
58   volume_id      = "${aws_ebs_volume.my-test-ebs.*.id[count.index]}"
59 }
60
61 data "template_file" "user-init" {
62   template = "${file("${path.module}/userdata.tpl")}"
63 }
64
65 resource "aws_cloudwatch_metric_alarm" "cpu-utilization" {
66   alarm_name         = "high-cpu-utilization-alarm"
67   comparison_operator = "GreaterThanOrEqualToThreshold"
68   evaluation_periods  = "2"
69   metric_name         = "CPUUtilization"
70   namespace           = "AWS/EC2"
71   period              = "120"
72   statistic            = "Average"
73   threshold            = "80"
74   alarm_description   = "This metric monitors ec2 cpu utilization"
75   alarm_actions        = ["${var.alarm_actions}"]
76
77   dimensions {
78     InstanceId = "${aws_instance.test_instance.*.id[count.index]}"
79   }
80 }
81
82 resource "aws_cloudwatch_metric_alarm" "instance-health-check" {
83   alarm_name         = "instance-health-check"
84   comparison_operator = "GreaterThanOrEqualToThreshold"
85   evaluation_periods  = "1"
86   metric_name         = "StatusCheckFailed"
87   namespace           = "AWS/EC2"
88   period              = "120"
89   statistic            = "Average"
90   threshold            = "1"
91   alarm_description   = "This metric monitors ec2 health status"
92   alarm_actions        = ["${var.alarm_actions}"]
93
94   dimensions {
95     InstanceId = "${aws_instance.test_instance.*.id[count.index]}"
96   }
97 }
98 }
```


ec2_with_cloudwatch_alarm.tf hosted with ❤️ by GitHub [view raw](#)

GitHub Link

100daysofdevops/100daysofdevops

Contribute to 100daysofdevops/100daysofdevops development by creating an account on GitHub.

github.com



Looking forward from you guys to join this journey and spend a minimum an hour every day for the next 100 days on DevOps work and post your progress using any of the below medium.

- Twitter: [@100daysofdevops](#) OR [@lakhera2015](#)
- Facebook: <https://www.facebook.com/groups/795382630808645/>
- Medium: <https://medium.com/@devopslearning>
- Slack: <https://devops-myworld.slack.com/messages/CF41EFG49/>
- GitHub Link:<https://github.com/100daysofdevops>

Reference

100 Days of DevOps — Day 0

D-day is just one day away and finally, this is a continuation of the post(I posted a month earlier)

medium.com



62

Q

62

Q

62

Q

Get an email whenever Prashant Lakhera publishes.

Your email

Subscribe

We couldn't process your request. Try again, or contact our support team.

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

More from Prashant Lakhera

AWS Community Builder, Ex-Redhat, Author, Blogger, YouTuber, RHCA, RHCD, RHCE, Docker Certified,4XAWS, CCNA, MCP, Certified Jenkins, Terraform Certified, IXGCP

Feb 27, 2019

100 Days of DevOps — Day 17- Creating EC2 Instance using Terraform

Check the updated 101 Days of DevOps Course Course Registration link: <https://www.101daysofdevops.com/register/> Course Link:...

AWS

5 min read

62

Q

62

Q

62

Q

Share your ideas with millions of readers. [Write on Medium](#)

Feb 26, 2019

100 Days of DevOps — Day 16- Building VPC using Terraform

Check the updated 101 Days of DevOps Course Course Registration link: <https://www.101daysofdevops.com/register/> Course Link:...

AWS

5 min read

62

Q

62

Q

62

Q

Feb 25, 2019

100 Days of DevOps — Day 15- Introduction to Terraform

Check the updated 101 Days of DevOps Course Course Registration link: <https://www.101daysofdevops.com/register/> Course Link:...

AWS

15 min read

62

Q

62

Q

62

Q

Feb 24, 2019

100 Days of DevOps — Day 14- How to automate the process of EBS Snapshot Creation

Check the updated 101 Days of DevOps Course Course Registration link: <https://www.101daysofdevops.com/register/> Course Link:...

AWS

4 min read

62

Q

62

Q

62

Q

Published in **FAUN Publication** · Feb 23, 2019

100 Days of DevOps — Day 13- How to stop/start EC2 instance on a scheduled basis to save cost

Check the updated 101 Days of DevOps Course Course Registration link: <https://www.101daysofdevops.com/register/> Course Link:...

AWS

5 min read

62

Q

62

Q

62

Q

Love podcasts or audiobooks? Learn on the go with our new app. [Try Knowable](#)