**Cost optimization -🡪 Low utilized amazon ec2 instances+**

Checks the Amazon Elastic Compute Cloud (Amazon EC2) instances that were running at any time during the last 14 days and alerts you if the daily CPU utilization was 10% or less and network I/O was 5 MB or less on 4 or more days. Running instances generate hourly usage charges. Although some scenarios can result in low utilization by design, you can often lower your costs by managing the number and size of your instances.

Estimated monthly savings are calculated by using the current usage rate for On-Demand Instances and the estimated number of days the instance might be underutilized. Actual savings will vary if you are using Reserved Instances or Spot Instances, or if the instance is not running for a full day. To get daily utilization data, download the report for this check.

**Cost optimization -🡪 underutilized EBS volumes**

Checks Amazon Elastic Block Store (Amazon EBS) volume configurations and warns when volumes appear to be underused. Charges begin when a volume is created. If a volume remains unattached or has very low write activity (excluding boot volumes) for a period of time, the volume is probably not being used.

**Cost Optimization -🡪 Unassociated IP Addressed**

Checks for Elastic IP addresses (EIPs) that are not associated with a running Amazon Elastic Compute Cloud (Amazon EC2) instance. EIPs are static IP addresses designed for dynamic cloud computing. Unlike traditional static IP addresses, EIPs can mask the failure of an instance or Availability Zone by remapping a public IP address to another instance in your account. A nominal charge is imposed for an EIP that is not associated with a running instance.

|  |  |  |
| --- | --- | --- |
| PublicIp | Domain | AllocationId |
| 13.229.28.248 | vpc | eipalloc-4918012c |

Following is output of api please check that.

Highlighted output is represent that ip address is not attached to any instance .we can show that as output in table fotmat

>aws ec2 describe-addresses

{

"Addresses": [

{

"PublicIp": "13.229.28.248",

"Domain": "vpc",

"AllocationId": "eipalloc-4918012c"

},

{

"Domain": "vpc",

"InstanceId": "i-04f3ca0752ca02f8f",

"NetworkInterfaceId": "eni-2a0b6d50",

"AssociationId": "eipassoc-9ff948f9",

"NetworkInterfaceOwnerId": "975255136272",

"PublicIp": "13.228.179.107",

"AllocationId": "eipalloc-6fffe10a",

"PrivateIpAddress": "10.31.0.181"

}

]}

Security

## Security-🡪 security groups – specified port are unrestricted

## Checks security groups for rules that allow unrestricted access (0.0.0.0/0) to specific ports. Unrestricted access increases opportunities for malicious activity (hacking, denial-of-service attacks, loss of data). The ports with highest risk are flagged red, and those with less risk are flagged yellow. Ports flagged green are typically used by applications that require unrestricted access, such as HTTP and SMTP. If you have intentionally configured your security groups in this manner, we recommend using additional security measures to secure your infrastructure (such as IP tables).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GroupName | VpcId | OwnerId | GroupId | IpPermissions | Security status |
| DB-SecurityGroup | vpc-075f2263 | 686263021012 | 91ef80e8 | Button to view details in next page as following table format | RED/YELLOW/GREEN  [Please check](https://cloudacademy.com/blog/aws-trusted-advisor/) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IpRanges | ToPort | FromPort | IpProtocol | Rule status |
| 0.0.0.0/0 | 81 | 81 | tcp | RED |

Please check the following output of security group api we will find out api from and display in table format .

aws ec2 describe-security-groups

{

"SecurityGroups": [

{

"IpPermissionsEgress": [

{

"IpProtocol": "-1",

"PrefixListIds": [],

"IpRanges": [

{

"CidrIp": "0.0.0.0/0"

}

],

"UserIdGroupPairs": [],

"Ipv6Ranges": []

}

],

"Description": "default VPC security group",

"IpPermissions": [

{

"IpProtocol": "-1",

"PrefixListIds": [],

"IpRanges": [],

"UserIdGroupPairs": [

{

"UserId": "686263021012",

"GroupId": "sg-26228956"

}

],

"Ipv6Ranges": []

}

],

"GroupName": "default",

"VpcId": "vpc-e1b6e498",

"OwnerId": "686263021012",

"GroupId": "sg-26228956"

},

{

"IpPermissionsEgress": [

{

"IpProtocol": "-1",

"PrefixListIds": [],

"IpRanges": [

{

"CidrIp": "0.0.0.0/0"

}

],

"UserIdGroupPairs": [],

"Ipv6Ranges": []

}

],

"Description": "default VPC security group",

"IpPermissions": [

{

"IpProtocol": "-1",

"PrefixListIds": [],

"IpRanges": [],

"UserIdGroupPairs": [

{

"UserId": "686263021012",

"GroupId": "sg-5b9af522"

}

],

"Ipv6Ranges": []

}

],

"GroupName": "default",

"VpcId": "vpc-075f2263",

"OwnerId": "686263021012",

"GroupId": "sg-5b9af522"

},

{

"IpPermissionsEgress": [

{

"IpProtocol": "-1",

"PrefixListIds": [],

"IpRanges": [

{

"CidrIp": "0.0.0.0/0"

}

],

"UserIdGroupPairs": [],

"Ipv6Ranges": []

}

],

"Description": "default VPC security group",

"IpPermissions": [

{

"IpProtocol": "-1",

"PrefixListIds": [],

"IpRanges": [],

"UserIdGroupPairs": [

{

"UserId": "686263021012",

"GroupId": "sg-71781408"

}

],

"Ipv6Ranges": []

}

],

"GroupName": "default",

"VpcId": "vpc-2d245949",

"OwnerId": "686263021012",

"GroupId": "sg-71781408"

},

{

"IpPermissionsEgress": [

{

"IpProtocol": "-1",

"PrefixListIds": [],

"IpRanges": [

{

"CidrIp": "0.0.0.0/0"

}

],

"UserIdGroupPairs": [],

"Ipv6Ranges": []

}

],

"Description": "This is secoupurity group for database",

"Tags": [

{

"Value": "RDS group",

"Key": "Name"

}

],

"IpPermissions": [

{

"PrefixListIds": [],

"FromPort": 3306,

"IpRanges": [],

"ToPort": 3306,

"IpProtocol": "tcp",

"UserIdGroupPairs": [

{

"UserId": "686263021012",

"GroupId": "sg-c4ec83bd"

}

],

"Ipv6Ranges": []

}

],

"GroupName": "DB-SecurityGroup",

"VpcId": "vpc-075f2263",

"OwnerId": "686263021012",

"GroupId": "sg-91ef80e8"

},

{

"IpPermissionsEgress": [

{

"IpProtocol": "-1",

"PrefixListIds": [],

"IpRanges": [

{

"CidrIp": "0.0.0.0/0"

}

],

"UserIdGroupPairs": [],

"Ipv6Ranges": []

}

],

"Description": "WEbServer-Sg created 2016-01-15T20:22:50.444+05:30",

"Tags": [

{

"Value": "web server",

"Key": "Name"

}

],

"IpPermissions": [

{

"PrefixListIds": [],

"FromPort": 81,

"IpRanges": [

{

"CidrIp": "0.0.0.0/0"

}

],

"ToPort": 81,

"IpProtocol": "tcp",

"UserIdGroupPairs": [],

"Ipv6Ranges": []

},

{

"PrefixListIds": [],

"FromPort": 80,

"IpRanges": [

{

"CidrIp": "0.0.0.0/0"

}

],

"ToPort": 80,

"IpProtocol": "tcp",

"UserIdGroupPairs": [],

"Ipv6Ranges": []

},

{

"PrefixListIds": [],

"FromPort": 8080,

"IpRanges": [

{

"CidrIp": "0.0.0.0/0"

}

],

"ToPort": 8080,

"IpProtocol": "tcp",

"UserIdGroupPairs": [],

"Ipv6Ranges": []

},

{

"PrefixListIds": [],

"FromPort": 3389,

"IpRanges": [

{

"CidrIp": "120.138.116.242/32"

},

{

"CidrIp": "115.118.17.185/32"

},

{

"CidrIp": "115.114.191.92/32"

},

{

"CidrIp": "115.114.191.80/28"

},

{

"CidrIp": "103.6.33.0/24"

},

{

"CidrIp": "117.233.43.29/32"

},

{

"CidrIp": "117.228.108.183/32"

},

{

"CidrIp": "117.233.1.58/32"

},

{

"CidrIp": "59.163.69.0/24"

},

{

"CidrIp": "123.201.241.77/32"

},

{

"CidrIp": "123.201.165.198/32"

},

{

"CidrIp": "117.233.2.90/32"

},

{

"CidrIp": "137.59.66.38/32"

},

{

"CidrIp": "121.244.211.20/32"

},

{

"CidrIp": "34.192.188.119/32"

},

{

"CidrIp": "0.0.0.0/0"

}

],

"ToPort": 3389,

"IpProtocol": "tcp",

"UserIdGroupPairs": [],

"Ipv6Ranges": []

}

],

"GroupName": "WEbServer-Sg",

"VpcId": "vpc-075f2263",

"OwnerId": "686263021012",

"GroupId": "sg-c4ec83bd"

}

]

}

## Security-🡪 IAM Use

Checks for your use of AWS Identity and Access Management (IAM). You can use IAM to create users, groups, and roles in AWS, and you can use permissions to control access to AWS resources.

Display fowling repot in table format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User name | Last login | Last service use | Last password changed | User created date |
| Saurabh\_p | Fri Feb 24 19:49:56 IST 2017 | S3 | Fri Feb 24 19:49:56 IST 2017 | Fri Feb 24 19:49:56 IST 2017 |

## Security-🡪 s3 Bucket permission

Checks buckets in Amazon Simple Storage Service (Amazon S3) that have open access permissions. Bucket permissions that grant List access to everyone can result in higher than expected charges if objects in the bucket are listed by unintended users at a high frequency. Bucket permissions that grant Upload/Delete access to everyone create potential security vulnerabilities by allowing anyone to add, modify, or remove items in a bucket. This check examines explicit bucket permissions, but it does not examine associated bucket policies that might override the bucket permissions.

Display report as following table format

**Total GB = 100 Gb Total cost = $50** (please calculate with formula)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bucket name | List permission | Public access | Object count | Created date | Total size | Remark |
| Saurabh\_bucket | No | No | 5 | Fri Feb 24 19:49:56 IST 2017 | 5 GB | Bucket is secure |

**Security-Amazon public snapshots**

Checks the permission settings for your Amazon Elastic Block Store (Amazon EBS) volume snapshots and alerts you if any snapshots are marked as public. When you make a snapshot public, you give all AWS accounts and users access to all the data on the snapshot. If you want to share a snapshot with particular users or accounts, mark the snapshot as private, and then specify the user or accounts you want to share the snapshot data with.

## 

## Performance

## Performance 🡪 High Utilization Amazon ec2 instances

## Checks the Amazon Elastic Compute Cloud (Amazon EC2) instances that were running at any time during the last 14 days and alerts you if the daily CPU utilization was more than 90% on 4 or more days. Consistent high utilization can indicate optimized, steady performance, but it can also indicate that an application does not have enough resources. To get daily CPU utilization data, download the report for this check

Display content in table format

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instance id | Instance name | Instance type | CPU utilization | Status | Running since | Launch date |
| i-s51sk3jd887255 | Production-web | T2.miro | 90 | running | 120 day | 22-august-2017 |

**Settings**

**Settings🡪 User management**

* + Super admin user can
  + Add user
  + Edit user
  + Delete user

Most important security aspect is user password information is stored in data base as encrypted format. And maintain activity of user like last login and

We want to store user information as following

* User name – unique
* Password – must be secure – atlist one species charater 8 length longs and one number
* User first name
* User last name
* Email id
* Phone number
* Role – super admin or admin

Following table will display to show user information

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| User name | First name | Last name | Email id | Phone no | role | Created on |  |
| Saurabh\_pawar | Saurabh | Pawar | [saurabhpawar@gmail.com](mailto:saurabhpawar@gmail.com) | 1234567890 | Admin | 22-august-2017 6:00 PM | Delete  And edit button |

**Settings🡪 SMTP settings**

**Display following form with add and delete functionality – user can add only one smtp setting**

* **SMTP server url**
* **SMTP user name**
* **SMTP password**
* **SMTP port**
* **Ssl – true or false**
* **Sender email id**

**Settings🡪 Schedulers display**

**Display in table format following things**

* Job name
* Job description
* Next job running time
* Force run job (button)

Please check <https://www.mkyong.com/jsf2/how-to-trigger-a-quartz-job-manually-jsf-2-example/>

**Settings 🡪 Amazon Credentials**

* Access key
* Secret key
* Configuration name --- just user friendly name
* Region name – can choose multiple regions like us-east-1 (north vetginiya) ,