# Create EC2 in AWS.

EC2 creation catalogue is used to provision instance in both Windows and Linux OS.Below are the features that can be applied in Windows and Linux.

Windows OS:

* Provisioning the instance in windows OS
* By default, instance type is General Purpose t2 Micro | 1vCPU/1 GiB
* Attach IAMInstanceprofile to the instance
* Terminate the instance using CLI or instance if DisableAPITermination is set to false.
* Enable monitoring and backup.
* Backup is been done by default backup policy (AWSBackupPolicyDefault).
* Attach maximum five numbers of volumes to the instance with disk size of (20,40,60,100).
* Volume can also be encrypted using KMS Key ID.
* Each drive should have unique name (C-Z).

Linux OS:

* Provisioning the instance in Linux OS
* By default, instance type is General Purpose t2 Micro | 1vCPU/1 GiB
* Attach IAMInstanceprofile to the instance
* Terminate the instance using CLI or instance if DisableAPITermination is set to false.
* Enable monitoring and backup.
* Backup is been done by default backup policy (AWSBackupPolicyDefault).
* Attach maximum five numbers of Volumes.
* Each Volume have their Disk size of (20,40,60,100) and maximum three of logical volume i.e mount points.
* Each Disk name are auto-generated and unique.

We have option to delete the disk on termination and also can be encrypted.The Volume types can be Standard,General Purpose and Provisioned IOPS.

## Service Now Workflow:

## Sample Payload

**Sample Input linux**:

{

"ScriptFiles": [

{

"Name": "EC2Creation.py",

"Type": "PY",

"Uri": "CE\_Core\_Templates/AWS/IaaS\_Automation/Compute Provisioning/Ec2InstanceCreation\_V2.py"

},

{

"Name": "EC2Creation.yml",

"Type": "CFT",

"Uri": "CE\_Core\_Templates/AWS/IaaS\_Automation/Compute Provisioning/EC2InstanceCreation.yml"

}

],

"ScriptPayload": {

"Region": "us-west-2",

"KmsKeyId": "",

"InstanceName": "CEDevFinEC2036",

"DeleteOnTermination": "false",

"DisableApiTermination": "false",

"EbsOptimized": "false",

"Encrypted": "false",

"IamInstanceProfile": "aws-elasticbeanstalk-ec2-role",

"InstanceInitiatedShutdownBehavior": "stop",

"InstanceType": "t2.micro",

"Iops": "",

"KeyName": "cedemo-oregon-keypair",

"Monitoring": "false",

"OperatingSystem": "AmazonLinux",

"SecurityGroupId": "sg-33d71174",

"SubnetId": "subnet-b94cbfe4",

"Tenancy": "default",

"UserData": "mkdir testttt\nmkdir test222",

"AvailabilityZone": "us-west-2c",

"VolumeType": "gp2",

"additionalDiskSize": [],

"additionalDiskName": [],

"additionalDiskDriveLetter": [],

"DiskPath": [],

"LinuxFormat": [

{

"DiskSize": "20",

"DiskName": "CEDevFinVOLUMENaN",

"Diskpath": "/dev/xvdf",

"PVName": "/dev/xvdf",

"VGName": "CEDevFinVGNaN",

"LVNames": [

"CEDevFinLV016",

"CEDevFinLV017",

"CEDevFinLV018"

],

"MountNames": [

"/test1",

"/test2",

"/test3"

],

"Lvsize": [

"5",

"5",

"5"

]

}

],

"StackName": "RITM0010341-EC2Creation1",

"Tags": [

{

"Key": "Owner",

"Value": "System Administrator"

},

{

"Key": "Project",

"Value": "221db0edc611228401760aec06c9d929"

},

{

"Key": "BusinessUnit",

"Value": "Finance"

},

{

"Key": "Application",

"Value": "apache linux den 200"

},

{

"Key": "Environment",

"Value": "Development"

},

{

"Key": "CreatedBy",

"Value": "CloudExponence"

},

{

"Key": "RequestID",

"Value": "RITM0010341"

}

]

}

}

**Sample Input Windows**

{

"ScriptFiles": [

{

"Name": "EC2Creation.py",

"Type": "PY",

"Uri": "CE\_Core\_Templates/AWS/IaaS\_Automation/Compute Provisioning/Ec2InstanceCreation\_V2.py"

},

{

"Name": "EC2Creation.yml",

"Type": "CFT",

"Uri": "CE\_Core\_Templates/AWS/IaaS\_Automation/Compute Provisioning/EC2InstanceCreation.yml"

}

],

"ScriptPayload": {

"Region": "us-west-2",

"KmsKeyId": "",

"InstanceName": "CEDevFinEC2018",

"DeleteOnTermination": "false",

"DisableApiTermination": "false",

"EbsOptimized": "false",

"Encrypted": "false",

"IamInstanceProfile": "aws-elasticbeanstalk-ec2-role",

"InstanceInitiatedShutdownBehavior": "stop",

"InstanceType": "t2.micro",

"KeyName": "cedemo-oregon-keypair",

"Monitoring": "false",

"OperatingSystem": "Windows",

"SecurityGroupId": "sg-33d71174",

"SubnetId": "subnet-b94cbfe4",

"Tenancy": "default",

"UserData": "New-Item -Path \"c:\\\" -Name \"logfiles\" -ItemType \"directory\"",

"AvailabilityZone": "us-west-2c",

"additionalDiskSize": [

"10",

"5",

"20"

],

"additionalDiskName": [

"vmnamef",

"vmnameg",

"vmnameh",

"vmnamei",

"vmnamej"

],

"additionalDiskDriveLetter": [

"F",

"G",

"H",

"",

""

],

"DiskPath": [

"/dev/sdf",

"/dev/sdg",

"/dev/sdh"

],

"VolumeType": "gp2",

"StackName": "RITM0010318-EC2Creation",

"Tags": [

{

"Key": "Owner",

"Value": "System Administrator"

},

{

"Key": "Project",

"Value": "221db0edc611228401760aec06c9d929"

},

{

"Key": "BusinessUnit",

"Value": "Finance"

},

{

"Key": "Application",

"Value": "apache linux den 200"

},

{

"Key": "Environment",

"Value": "Development"

},

{

"Key": "CreatedBy",

"Value": "CloudExponence"

},

{

"Key": "RequestID",

"Value": "RITM0010318"

}

]

}

}

### Developer Notes:

EC2 Creation – Form Design

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Validation if any** |
| Tier | Drop Down | Application,Databases,Shared Services,Web | None |
| Instance type | Drop Down | If t2micro,t2small etc should be got a from a table | None |
| OS | Drop Down | Amazon Linux,Windows | None |
| Role | Drop Down | All roles to be displayed | None |
| EC2 key pair | Drop Down | All key pairs of that region | None |
| Instance shutdown behavior | Drop Down | Stop/Terminate | None |
| Advanced Instance type | Drop Down | More options got from table which has field as advanced | None |
| Tenancy | Drop Down | Default,Dedicated,Host | None |
| Disable APi Termincation | Check box | To check if the instance can be terminated from api calls | None |
| Enable Monitoring | Check box | If cloud watch monitoring to be enabled for the instance | None |
| Enable Back up | Check box | If the instance has to be applied to the awsbackup policies.(when clicked the default policy will be shown as read only). This is got from the backup table for the selected account and region | None |
| Disk Name | Single line text | /dev/sdc etc populated on load. Only for linux | None as read only |
| Disk Size | Drop down | 20,40,60,100 .Only for linux | None |
| Mount Point Count | Drop down | 1,2,3. Only for linux | None |
| Mount Point | Single line text | Example /data1 .Only for linux | Path validation. Unique mount point validation for all mount points |
| Size(GB) | Single line text | Integer values. Only for linux | Integer validation, Sum of all mount points should be lesser than disk size |
| Encrypted | Check box | If the volume has to be encypted or not | None |
| KMS key id | Drop down | All kms key id of the region | None |
| Volume type | Select box | General Purpose,Provisioned IOPS,Standard | None |
| VM Drive | Drop Down | Alphabets F to Z .Only for Windows | Drives should not be repeated |
| VM Disk Size | Drop Down | 20,30,60,100. Only for Windows | None |

Internal Mapping in Form

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Mapping** |
| AZ id | String | AZ | Got from subnet table based on region ,subscription,environment and tier |
| Subnet id | String | Id of subnet | Got from subnet table based on region ,subscription,environment and tier |
| VPC id | String | VPC | Got from subnet table based on region ,subscription,environment and tier |
| Security group id | String | SG id | Got from subnet table based on region ,subscription,environment and tier |
| Back up policy name | String | Policy name | Got from backup table for the subscription and region |
| EC2 name Vgname,lvname, | String | Names of ec2,volume group and logical volume | Got from name pattern table |
| Iops | String | Iops to be passed only if volume type is provisioned iops | Disk size \* 50 |

Automation Workflow:

The “ScriptFiles” section contains the scripts location in the github repo and ScriptPayload” contains the list of inputs to be passed.

The ResourceDeploymentStepapigateway invokes ResourceDeploymentSingleStepFunction.

The following operations happen in the “ResourceDeploymentSingle”stepfunction:

1. The “ResourceDeploymentStep” lambda is invoked .
2. Lambda will check the input format of payload and verifies it. It also get the credentials through the role associated to it for the below executions.
3. It will get the git repo credentials from the SSM parameter store and then download the EC2Creation.py and EC2Creation.yml files in s3.
4. Then the inputs will be passed to the python file where it will trigger the CloudFormation Template file from S3 bucket if the stack has to be created. If there is no stack creation, python file itself will do the required activity.
5. The final activity of the step function is to notify success message to servicenow if the stack is successfully created.
6. If there is no stack creation involved, it will directly notify the servicenow after the python scripts successfully executes

Error Handling:

1. Step function will check if the stack is successfully created. It will notify service now of the status of stack creation if error
2. If there is no stack creation, it will notify servicenow if there are any error from python scripts.
3. Any other runtime error from step function /lambda is also notified to servicenow.