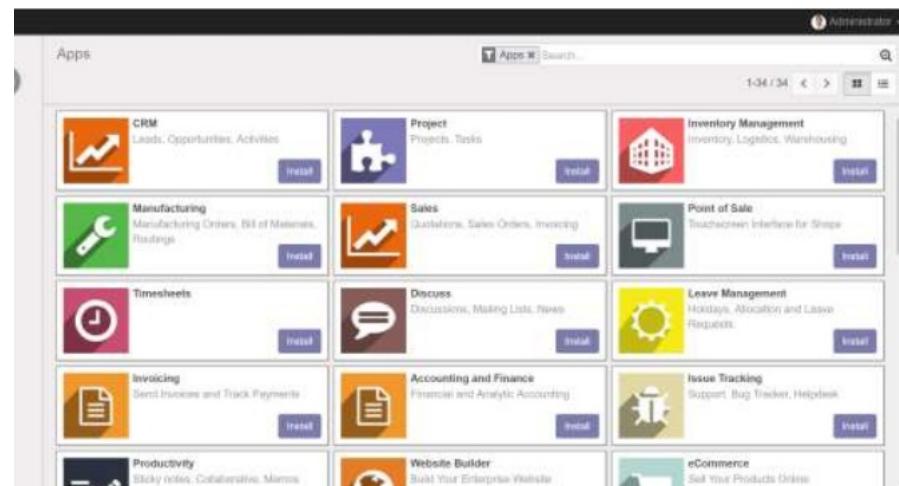


If everything goes well, you will see the screen below:

[Go to Odoo](#) | [Add preferences anytime.](#)

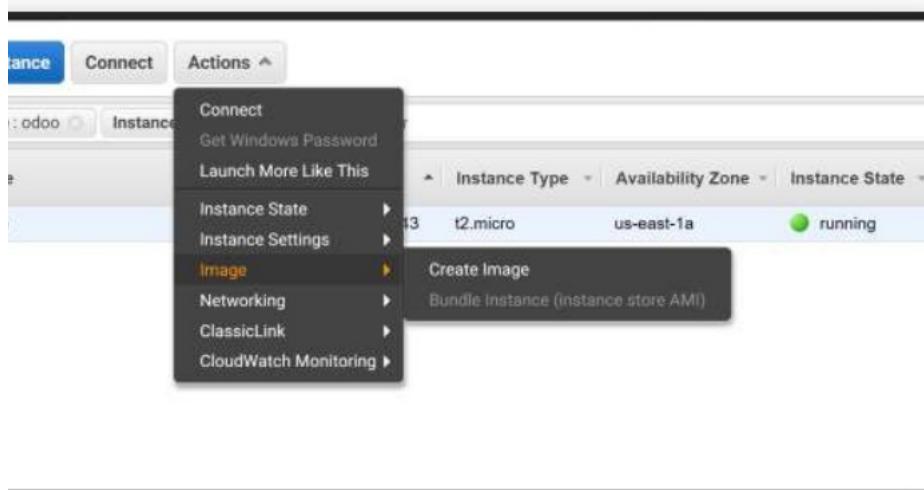


## MI images

With the steps taken above, we can now create our Odoo AMI. This machine image will allow us to create new Odoo instances.

machine, go to the EC2 dashboard > Select your newly created Odoo

Action > Image > Create Image as shown below



#### [Q&A IT Co. Ltd.](#)

missing crucial parts... first is static data such as session information, the best is to

re and the session storage shareable within your another more elegant solution is to use EFS just

the auto scaling group and the newly created AMI to launch a second the other Availability Zone.

with pgpool for example?

do you think it's time to change my mind?

## ng configuration

“Auto Scaling helps you maintain application availability and allows you Amazon EC2 capacity up or down automatically according to you define. You can use Auto Scaling to help ensure that you are the desired number of Amazon EC2 instances. Auto Scaling can also increase the number of Amazon EC2 instances during demand maintain performance and decrease capacity during lulls to reduce Scaling is well suited both for applications that have stable demand that experience hourly, daily, or weekly variability in usage” (ref: [amazon.com/autoscaling/](https://aws.amazon.com/autoscaling/))

[stemAdministrator upwork.com](#)

auto scaling group

Auto scaling >> launch configuration

“Create Auto Scaling group” button

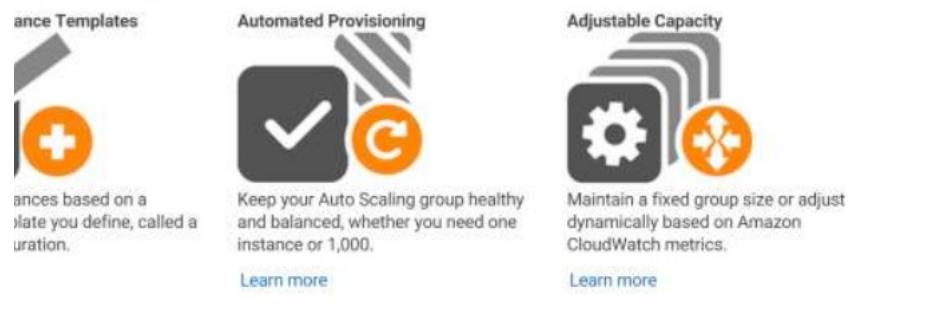
to Auto Scaling

uto Scaling to manage Amazon EC2 capacity automatically, maintain the right number of instances for your application, itly group of instances, and scale it according to your needs.

to Scaling group

your Auto Scaling groups in a different region, select your region from the navigation bar.

## Auto Scaling

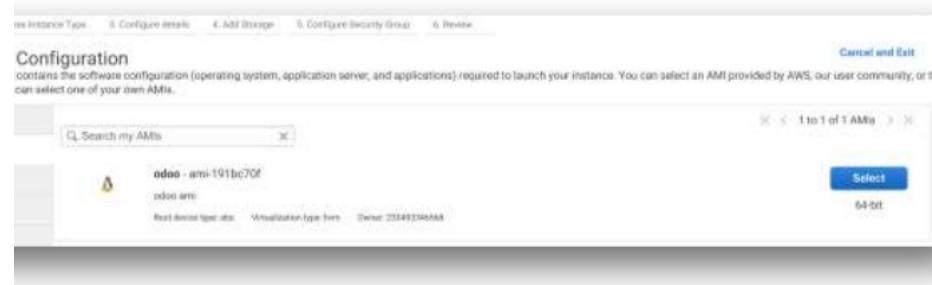


On the “create launch configuration” button to create a launch configuration group. In the next screen, select the tab called “My AMIs”.

the “Create Auto Scaling Group” button.



I will select the Odoo AMI that we have previously taken the snapshot of.



Instance type selection step, we will select an instance type t2.micro. Our main goal is to save cost and take advantage of the highly available scalable AWS infrastructures. Let's click on the “Next: Configuration

ation software that is suitable for SMEs. Small because it does not give them a highly available, an Odoo deployment that costs less than scalable. All this, because of its cloud possible. Objectives: Being able to build a highly own as OpenERP and before that, TinyERP) is a accounting, manufacturing, purchasing, Odoo version 10 community edition for this Odoo implementers decided to only show few and the general networking configuration that cause there are plenty of literature available for solution architecture to be simple and readable. is the Auto scaling viewpoint diagram. This diagram. A good solution architecture design viewpoint.

ware using the AWS auto scaling capabilities to very rapidly. The system is fault tolerant client. It is cost-efficient because provisioning the scope of this document we will skip the - AWS: to create an AWS account go to am/ - AWS VPC: read more here WS S3 and AWS EC2 policies VPC • set up es RDS on Multiple AZ • Create the Odoo user Odoo port rewrite on Nginx Auto Scaling • step 53 • Register a domain • Link the ELB to the

asticloadbalancing/ - AWS Route 53 Odoo AMI What is an AMI? According to achieve within the Amazon Elastic Compute apt install awscli -y InstallOdoo wget -O /sources.list.d/odoo.list apt-get update && apt the commands below but replace db\_host with = admin db\_host= your.aws.rds.endpoint.url ;

ve Postgres SQL Now that we have configured \$sql -y service postgresql stop apt-get update -y ancing, media streaming, and more. It started out C can also function as a proxy server for email w.nginx.com/resources/glossary/nginx/ To you install Odoo, it runs on port 8069 by default. pe a weblink. Moreover, it is more convenient. ind: But before this let's stop Nginx service

mands below Create the Odoo file sudo nano original port that is port 8069 and the redirect port IP \$remote\_addr; proxy\_set\_header Host LI or command line interface. Ctrl + o and enter ns a file called "default". We need to delete it date command apt-get update -y too Initial deployment In the previous section, cess the Odoo application through the browser. everything goes well, you will see the screen

o with the demo data.

allow us to create new Odoo instances. To create Image as shown below We will use the auto

to scale your Amazon EC2 capacity up or your desired number of Amazon EC2 instances. in performance and decrease capacity during lulls hourly, daily, or weekly variability in usage" guration Click on the “Create AutoScaling ext screen, select the tab called “My AMIs”. on step, we will select an instance type t2.micro ures. Let's click on the “Next: Configuration

the next step and then select your security guration” button. this section, such as configure the notification r this simple demo. certain criteria. Here we have decided to increase eis below 60% of the CPU usage. group will a new instance. This is the mark the

:ontinue

[Choose Instance Type](#) [3. Configure details](#) [4. Add Storage](#) [5. Configure Security Group](#) [6. Review](#)

**Configuration**  
The selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Type	Type	vCPUs	Memory (GB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
purpose	t2.micro	1	1	EBS only	-	Low to Moderate

[Find more options](#) [Cancel](#) [Previous](#) [Next: Configure details](#)



I name our launch configuration and select an AWS IAM role that have S3. Continue to the next step and then select your security group.

[2. Choose Instance Type](#) [3. Configure details](#) [4. Add Storage](#) [5. Configure Security Group](#) [6. Review](#)

**Launch Configuration**

Name [i](#)

Purchasing option [i](#)  Request Spot Instances

IAM role [i](#)

Monitoring [i](#)  Enable CloudWatch detailed monitoring  
[Learn more](#)

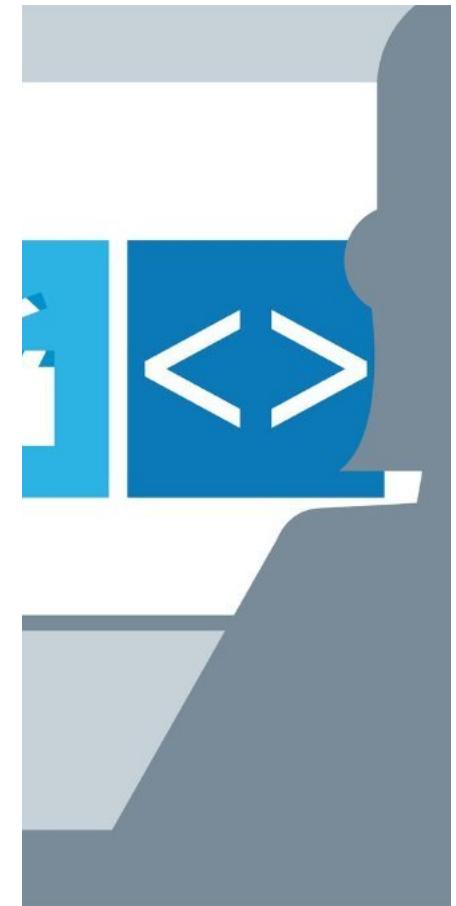


urity group that allows a public internet access on port 80, the. clicked at “Create launch configuration” button.

Web-DMZ vpo-27c0a641 Internet connection [Copy to new](#)

1467031 Selected security groups: sg-4d467031.

Protocol	Port Range	Source
TCP	80	0.0.0.0/0
TCP	22	50.20.98.2/32
Any	Any	Any

[Cancel](#) [Previous](#) [Review](#)

Edit instance type

Category	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
	EBS only	-	Low to Moderate

[Edit details](#)

[Edit storage](#)

[Edit security groups](#)

[Cancel](#) [Previous](#) [Create launch configuration](#)

Figure details 4. Add Storage 5. Configure Security Group 6. Review

Select an existing key pair or create a new key pair [X](#)

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

[Choose an existing key pair](#)

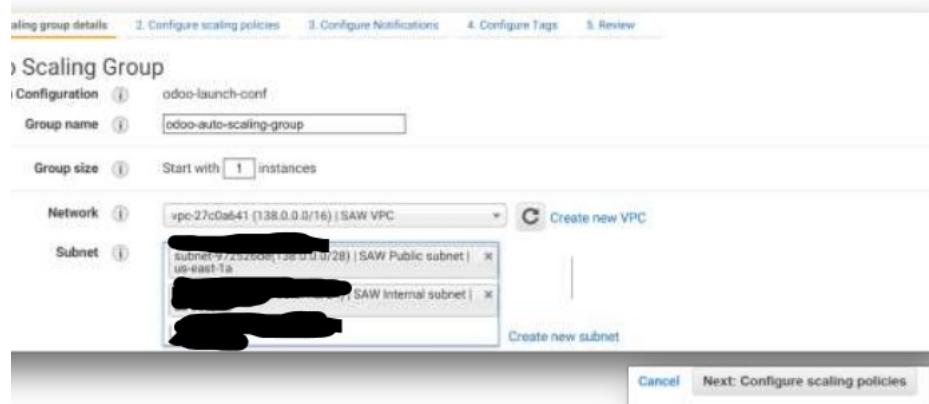
[Select a key pair](#)

I acknowledge that I have access to the selected private key file (Linuxkey.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) [Create launch configuration](#)

Step 5 is to create an Auto Scaling group and configure the auto scaling. There are other steps, in this section, such as configure the notification.

Configure the tags before we can review everything. But we will skip as they are not necessary for this simple demo.



Scaling policies are instructions given to the Auto Scaling group to launch instances based on certain criteria. Here we have decided to increase the number of instances by 1 one whenever the average CPU usage reaches 60% and decreases by 1 when that average is below 60% of the CPU usage.

[1. Auto Scaling Group details](#) [2. Configure scaling policies](#) [3. Configure Notifications](#) [4. Configure Tags](#) [5. Review](#)

### Auto Scaling Group

Scaling policies let you want to adjust the size (number of instances) of your group automatically. A scaling policy is a set of instructions for making such adjustments in response to the existing group size, or you can set the group to an exact size. When the alarm triggers, it will execute the policy and adjust the size of your group accordingly. Learn more about scaling policies.

At its initial size, the group has 1 instance. You can choose to increase or decrease the capacity of this group.

Group Size

Name:  When:  Add new alarm

breaches the alarm threshold: CPUUtilization <= 60 for 300 seconds for the metric dimensions AutoScalingGroupName = saw-odoo-auto-scale-group

Action:  1 instances when 60 >= CPUUtilization > -infinity Add step ⓘ

Time needed:  seconds to warm up after each step

Scaling policy ⓘ

Group Size

Name:  When:  Add new alarm

breaches the alarm threshold: CPUUtilization <= 60 for 300 seconds for the metric dimensions AutoScalingGroupName = saw-odoo-auto-scale-group

Action:  1 instances when 60 >= CPUUtilization > -infinity Add step ⓘ

Cancel Previous Review Next: Configure Notifications

[Create Auto Scaling group](#)

[Edit details](#)

[Edit scaling policies](#)

[Edit notifications](#)

[Edit tags](#)

[Create Auto Scaling group](#)