SUSE Linux Enterprise Server 12 SP5

SUSE Linux Enterprise in the Public Cloud

Amazon EC2 Container Service

Amazon EC2 Container Service (ECS) (https://aws.amazon.com/ecs/) is a container management service that provides a set of APIs for scheduling container workloads across EC2 clusters.

SUSE Linux Enterprise 12 images with the **ecs** TLA in the image name are designed to integrate with the Amazon EC2 Container service. The image contains an agent that interacts with Docker Open Source Engine to start/stop new containers and gather information about running containers. The agent is fully integrated with SUSE Linux Enterprise.

Publication Date: November 07, 2024

Contents

1 Cluster setup 2

2 ECS instance role 2

3 Attach SUSE Linux Enterprise Server ECS instances to the cluster 3

4 Additional Considerations 5

5 Using SUSE Linux Enterprise Server Containers 6

6 Docker Image Storage 6

7 Cleaning Up 7

1 Cluster setup

Creating a new ECS cluster in Amazon is accomplished by using the First Run Wizard. (https://console.aws.amazon.com/ecs/home#/firstRun) Alternatively you may use the `aws` command line tool to create the cluster, refer to create (http://docs.aws.amazon.com/cli/latest/reference/ecs/create-cluster.html) for details. In the First Run Wizard select *Custom* if you want to create a setup unique to your own needs. The *Amazon ECS sample* selection creates a pre-defined web application setup, that may be customized. By default the First Run Wizard selects the Amazon ECS AMI, we will change this later.

Note

Note that using the First Run Wizard also creates an autoscaling group. This implies that at least one instance (default setting) is always running and a new instance will be started by the autoscaling group if the running instance should be stopped or terminated.

We recommend retaining the default cluster name **default**. When the cluster is named `default` an ECS instance will automatically attach itself to the cluster without additional configuration steps. Selecting a custom name for the cluster requires the instance to be configured as outlined in *Section 3.1, "Alternative cluster name"*.

2 ECS instance role

Use of the Amazon EC2 Container Service requires permissions that are not enabled by default in the Identity and Access Management (IAM) functionality. Therefore it is necessary to create a policy that provides users that are intended to use ECS with the proper permissions. We recommend the following policy:

```
{
      "Version": "2012-10-17",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  "ecs:CreateCluster",
                  "ecs:DeregisterContainerInstance",
                  "ecs:DiscoverPollEndpoint",
                  "ecs:Poll",
                  "ecs:RegisterContainerInstance",
```

```
                "ecs:Submit*",
                "ecs:StartTask",
                "ecs:DescribeTasks",
                "ecs:StopTask",
                "ecs:RegisterTaskDefinition",
                "ecs:DescribeTaskDefinition",
                "ecs:DescribeContainerInstances",
                "ecs:StartTelemetrySession"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

For additional information about IAM policies, see the Amazon documentation: IAM Policies (http://docs.aws.amazon.com/AmazonECS/latest/developerguide/IAM_policies.html)

Once the policy is created attach it to a role named **ecsInstanceRole**. Details about ECS specific policies and roles can be found in the Amazon documentation: IAM Role (http://docs.aws.amazon.com/AmazonECS/latest/developerguide/instance_IAM_role.html)

Note

Note that the policy given above is more permissive than the policy example provided in the Amazon documentation.

3 Attach SUSE Linux Enterprise Server ECS instances to the cluster

Once the cluster is up and running and your AWS account provides a permissive IAM role for ECS, you can continue to run ECS instances as follows:

1. Find the latest SUSE Linux Enterprise Server ECS ami using pint (https://www.suse.com/communities/conversations/tag/pint)

   ```
   $ pint amazon images --active --filter 'name~ecs'
   ```

2. Run as many instances of the above ami as required for your cluster. Remember to select the ECS role when launching the instance.

You may use the Amazon EC2 Web Console to launch instances. The SUSE Linux Enterprise Server ECS images are found in the so called General Catalog. From the **Instances** view select *Launch Instance*. In the left column select the *Community AMIs* tab. Check the *SUSE Linux* box on the left and enter **ecs** into the search box. When configuring the instance parameters in the **Configure Instance Details** view select the *ecsInstanceRole* as the IAM role. Then launch the instance.
Alternatively you can use the `aws` command line tool to run instances, see run-instances (http://docs.aws.amazon.com/cli/latest/reference/ec2/run-instances.html) for details.

```
$ aws ec2 run-instances \
           --image-id IMAGE_ID_SEE_PINT_OUTPUT \
           --region EC2_REGION_NAME \
           --key-name SSH_KEY_PAIR_NAME \
           --instance-type DESIRED_INSTANCE_TYPE \
           --subnet-id SUBNET_ID \
           --iam-instance-profile 'Arn=PROFILE_PATH'
```

The *PROFILE_PATH* may be obtained by querying the IAM service.

```
$ aws iam list-instance-profiles
```

The *SUBNET_ID* will depend on your network configuration. If you created your setup using the First Run Wizard you will most likely want the instance to run in the subnet created by the wizard. Thus, you can obtain the value by inspecting the properties of the running ECS Instance.
Once the instance is running and if your cluster is named `default` you can confirm that the instance has attached itself to the cluster as follows:

```
$ aws ecs describe-clusters
```

The `registeredContainerInstancesCount` entry in the output provides the number of instances considered to be part of the cluster. If you used the First Run Wizard to create the cluster the number should be at least 2 after you launch the SUSE Linux Enterprise Server ECS instance.

3. If you used the First Run Wizard you already have an Auto Scaling (https://aws.amazon.com/autoscaling/) group. You can add (http://docs.aws.amazon.com/cli/latest/reference/autoscaling/attach-instances.html) the running SUSE Linux Enterprise Server instance to the existing group, or simply remove the default group. Removing the group

created by the ECS First Run Wizard will also terminate the ECS Amazon AMI and you will be left with the running SUSE Linux Enterprise Server ECS optimized instance launched above.

The autoscaling feature provides parallel functionality that is not strictly necessary for operating container instances. The First Run Wizard creates an autoscaling group as a good practice to ensure that at least always 1 container host is running.

3.1 Alternative cluster name

If you chose a name other than **default** for your cluster it is necessary to configure the instance to attach itself to this cluster. The configuration consists of setting the **ECS_CLUSTER** variable as follows:

1. Login to the running ECS instance via ssh.

2. Edit the environment file `/etc/ecs/ecs.config` and add an entry of the form:
   ECS_CLUSTER = name
   The amazon ECS init launcher will read in the environment file and populate the information into the environment prior to starting the ECS agent. A complete list of configuration options can be found at: Agent Configuration Variables (https://github.com/aws/amazon-ecs-agent#environment-variables)

3. Restart the service with:

   ```
   $ sudo systemctl restart amazon-ecs
   ```

   The current instance will be detached from the default cluster and will be re-attached to the configured cluster name.
   Watching `/var/log/ecs/ecs-agent.log.*` will tell you if the expected cluster is used.

4 Additional Considerations

The integrated ECS service requires access to Amazon S3 to download a pre-built container. This implies that if you start an ECS instance that has no route to reach Amazon S3 the initialization will fail and the container host instance will not be able to attach itself to your cluster. Should

you end up in a situation where the instance does not attach itself to your **default** cluster, you can modify your network configuration to ensure a route to Amazon S3 and then simply log into your container instance as start the service as follows:

```
$ systemctl start amazon-ecs.service
```

5 Using SUSE Linux Enterprise Server Containers

SUSE Linux Enterprise Server Docker images are not available on Docker Hub. However, SUSE provides pre-built Docker images as packages in the SLE-Module-Containers12-Updates repository. Given that your launched instance has a route to the Internet and that the traffic originates with an IP address assigned by EC2 the instance will automatically register with an update server and the SLE-Module-Containers12-Updates repository will be configured. Once configured, installing a SUSE Linux Enterprise Server Docker image is simple:

```
$ zypper se docker-image
```

Install the image you are interested in, the SUSE Linux Enterprise Server 12 image for example:

```
$ zypper in sles12-docker-image
```

The SUSE Linux Enterprise Server license does not allow you to push this image to a public Docker Registry. However you can use the Amazon-hosted private Container Registry (https://aws.amazon.com/ecr/) to store SUSE Linux Enterprise Server containers.

For additional details about SUSE Linux Enterprise Server Docker images see Quick Start Guide (https://documentation.suse.com/sles/html/SLES-all/docker-building-images.html#Customizing-Pre-build-Images).

6 Docker Image Storage

The SUSE Linux Enterprise Server ECS image uses Btrfs as the storage backend for Docker image storage. The image storage is self contained in the `/var/lib/docker-storage.btrfs` file that is mounted to `/var/lib/docker`. The image storage is configured to be 100 GB in size. This is thinly provisioned and thus does not actually take up 100 GB of space on the root file system. It is the users responsibility to start an instance with a sufficiently large root volume to store the Docker images.

If you store a large number of Docker images or you want to safeguard your local Docker images from accidental deletion you may consider to place the image storage onto an external Elastic Block Store (https://aws.amazon.com/ebs/) device. If you format the device with Btrfs no changes to the Docker configuration are required.

Note

The device must be mounted to `/var/lib/docker`.

If you choose another storage backend you will need to modify `/etc/sysconfig/docker` to pass the appropriate configuration options to Docker Open Source Engine at start up. For information about storage configuration and storage related options, see the *Docker User Guide* at https://docs.docker.com/engine/userguide/storagedriver/imagesandcontainers/.

7 Cleaning Up

If you decide that you want to clean up and shut everything down there are a few additional steps to take when compared to terminating a system that is not associated with a cluster.

To delete a cluster the tasks count of the tasks associated with the cluster must be zero.

```
$ aws ecs update-service \
     --service TASK_NAME \
     --desired-count 0
```

Where *`TASK_NAME`* is the name of the task associated with the cluster. If you used the Quick Launch Wizard and selected *Amazon ECS sample* the *`TASK_NAME`* is `sample-webapp`.

After setting the task/service count to zero as shown above you can delete the task/service as follows:

```
$ aws ecs delete-service --service TASK_NAME
```

As mentioned previously, the First Run Wizard also creates an autoscaling group. This group also needs to be deleted.

Note

Any running instances associated with the autoscaling group will be terminated when the autoscaling group is deleted.

```
$ aws autoscaling delete-auto-scaling-group \
     --auto-scaling-group-name GROUP_NAME
```

Now you are ready to delete the cluster.

```
$ aws ecs delete-cluster --cluster CLUSTER_NAME
```