When using the `registry://<value-name>` format, the credential spec is read from the Windows registry on the daemon's host. The specified registry value must be located in:

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Virtualization\Containers\CredentialSp
ecs
```

## Create services using templates

You can use templates for some flags of `service create`, using the syntax provided by the Go's text/template (http://golang.org/pkg/text/template/) package.

The supported flags are the following :

- `--hostname`
- `--mount`
- `--env`

Valid placeholders for the Go template are listed below:

| Placeholder | Description |
| --- | --- |
| .Service.ID | Service ID |
| .Service.Name | Service name |
| .Service.Labels | Service labels |
| .Node.ID | Node ID |
| .Node.Hostname | Node Hostname |
| .Task.ID | Task ID |
| .Task.Name | Task name |
| .Task.Slot | Task slot |

### TEMPLATE EXAMPLE

In this example, we are going to set the template of the created containers based on the service's name, the node's ID and hostname where it sits.

```
$ docker service create --name hosttempl \
                        --hostname="{{.Node.Hostname}}-{{.Node.ID}}-{{.Service.Name}}"\
                         busybox top


va8ew30grofhjoychbr6iot8c

$ docker service ps va8ew30grofhjoychbr6iot8c

ID              NAME            IMAGE
                                NODE            DESIRED STATE   CURRENT STATE               ERRO
R   PORTS
wo41w8hg8qan  hosttempl.1  busybox:latest@sha256:29f5d56d12684887bdfa50dcd29fc31eea4aaf4
ad3bec43daf19026a7ce69912  2e7a8a9c4da2  Running         Running about a minute ago

$ docker inspect --format="{{.Config.Hostname}}" 2e7a8a9c4da2-wo41w8hg8qanxwjwsg4kxpprj-
hosttempl

x3ti0erg11rjpg64m75kej2mz-hosttempl
```

## Specify isolation mode (Windows)

By default, tasks scheduled on Windows nodes are run using the default isolation mode configured for this particular node. To force a specific isolation mode, you can use the `--isolation` flag:

```
$ docker service create --name myservice --isolation=process microsoft/nanoserver
```

Supported isolation modes on Windows are:

- `default` : use default settings specified on the node running the task
- `process` : use process isolation (Windows server only)
- `hyperv` : use Hyper-V isolation

## Create services requesting Generic Resources

You can narrow the kind of nodes your task can land on through the using the `--generic-resource` flag (if the nodes advertise these resources):

```
$ docker service create --name cuda \
                        --generic-resource "NVIDIA-GPU=2" \
                        --generic-resource "SSD=1" \
                        nvidia/cuda
```