

Spot Market

Jem Camba - 2018-10-19 - in Deployment Options

Bid on spare Packet server capacity at greatly reduced rates.

Overview

Packet's Spot Market allows users to bid on spare server capacity at reduced rates.

In exchange, you give Packet the right to revoke any instance with only a two minute warning (you may also choose to convert a revoked instance to list pricing when the two minute warning is given via the API).

If you have portable workload, the Spot Market is an excellent tool to help you reduce the cost of handling your workload.

How does the Spot Market work?

In short: the spot market is a marketplace with constantly changing inventory. You can place bids on this inventory by defining a variety of factors such as the maximum price you are willing to pay for a particular resource.

For example, you can provision five instances and pass the `spot_price_max` parameter with the maximum price you want to pay per hour per instance. If your price is equal to or greater than the current spot price for the given facility / plan, then the instances will be created for you.

These instances are volatile, which means that they can be revoked by Packet at any time if another user bids with a higher price.

Spot Instance Creation

Click here to view the [relevant API docs](#).

This works much like creating a traditional instance. The only difference in the API call is that we need two additional parameters:

- **termination_time** - An optional fixed date and time [UTC] in the future at which to terminate the instance. This does not guarantee your instance will run until that time as we may need to revoke the instance. By default the system will terminate an instance at 120 seconds if the resource is required.
- **spot_price_max** - The maximum price you are willing to pay for the instance, per hour. This should be greater or equal than the current spot price for the given facility and plan.

Here is an example in JSON format of making a spot instance request:

```
{
  "hostname": "spot-instance",
  "plan": "baremetal_1",
  "operating_system": "coreos_stable",
  "facility": "ewr1",
  "spot_instance": true,
  "billing_cycle": "hourly",
  "termination_time": "2017-07-01T00:00:00.000Z",
  "spot_price_max": 0.3
}
```

*Please Note: Date/Time is ISO8601. You can calculate the termination date/time by:
TZ=UTC date --date='+1 hour' --iso-8601=seconds*

Persistent Spot Market

Persistent Spot Market is the ability to create spot instances whenever the market price drops below a certain threshold without having to query the API constantly and issue device creation requests.

API Examples of Persistent Spot Market

Create:

POST /projects/acc88c1a-1ec3-4a9d-86bc-8febab4ef45b/spot-market-requests with the payload:

```
""
{
  "devices_max": 10,
  "devices_min": 8,
  "max_bid_price": 2,
```

```
"instance_parameters": {
  "hostname": "test.server.com",
  "description": "This is my test server. It's for testing.",
  "plan": "baremetal_1",
  "operating_system": "coreos_stable",
  "facility": "abc1",
  "billing_cycle": "hourly",
  "tags": ["abc1", "abc2"]
}
}
```

This should prompt a 200 response.

Delete:

```
DELETE /spot-market-requests/f01b4bb1-f93f-46d2-ab44-
e1709878208b?force_termination=true
```

Additional parameters:

```
devices_min = 5
devices_max = 10
facilities [array of facility codes or ids]
max_bid_price
device-specific parameters:planoperating
systemuserdatapublic_ipv4_subnet_size_ssh_keysoperating_systemhostnames
end_at
```

Spot Market Termination

Ascertaining when an instance will be terminated is not visible through our customer portal. Instead, you will need to curl against our API to get this information.

- From the API:

```
curl -X GET --header 'Accept: application/json' --header 'X-Auth-Token: token'
'https://api.packet.net/projects/projectUUID/devices' | jq '.termination_time'
```

The above will output the time when an instance will be revoked/terminated. If the response is `null`, it indicates that no termination time has been set.

- From within an instance:

```
curl -s https://metadata.packet.net/metadata | jq '.termination_date'
```

Same as before, the response will be a termination date/time or it will be null (e.g. a time/date has not yet been set for termination of your instance).

Please Note: *The default termination is ~120 seconds. This terminations occurs when your [max_spot_price](#) is below the current spot price max or when we are requiring more devices to be free for on-demand and your device has the lowest price of all the spot instances.*

Current and Historical Market Pricing

The Current Market Price (“CMP”) reflects the price you need to specify to deploy a single instance at this point in time.

If you are looking to deploy more than one spot instance, we encourage you to submit lower-than-market bids, as these bids can still be fulfilled via a device batch request with >1 quantity.

If the listed spot market price is 10x the standard price for a server of that type, then we are out of capacity for spot market instances in that location.

How to query pricing:

- Current Prices: **REQUEST:** [/market/spot/prices](#) will provide the current prices for each Packet facility and plan.
- History: **GET** [/market/spot/prices/history?plan&facility](#) will allow you to check our prices up to 90 days in the past for a given plan & facility.

Bidding Strategies

While your specific bidding strategy is obviously entirely up to you, most users adopt one of these 4 bidding strategies based on their use case.

#1. Strictly Optimize Cost - Sacrifice availability for the deepest discount. This is great for long-running workloads that have flexible end dates.

- Use Case: Research-style tasks
- Suggested Bid: near minimum

#2. Cost/Availability Balance - Slightly higher price than strictly cost optimized, but allows you to increase the likelihood of getting and keeping an instance for longer periods of time.

- Use Case: Batch jobs that can handle some amount of reclamation

- Suggested Bid: 10-20% above minimum

#3. Bid On-Demand Price - Receive a discount anytime the spot price is lower than on-demand, immediately switching to on-demand when it goes higher.

- Use Case: Fully cloud native applications capable of easily moving workloads around
- Suggested Bid: on-demand price

#4. Optimize Continuity - Bid very high to ensure availability and continuity.

- Use Case: Accept some periods of higher-than-market prices in return for uptime continuity
- Suggested Bid: 2-3x on-demand price

External Resources

- [Terraform Kubernetes Packet provider workaround](#)

Tags

deploy

spot market