Microsoft

Getting Started Guide:
Deploying Puppet Enterprise in Microsoft Azure

puppet labs

Getting Started Guide:
Deploying Puppet Enterprise in Microsoft Azure

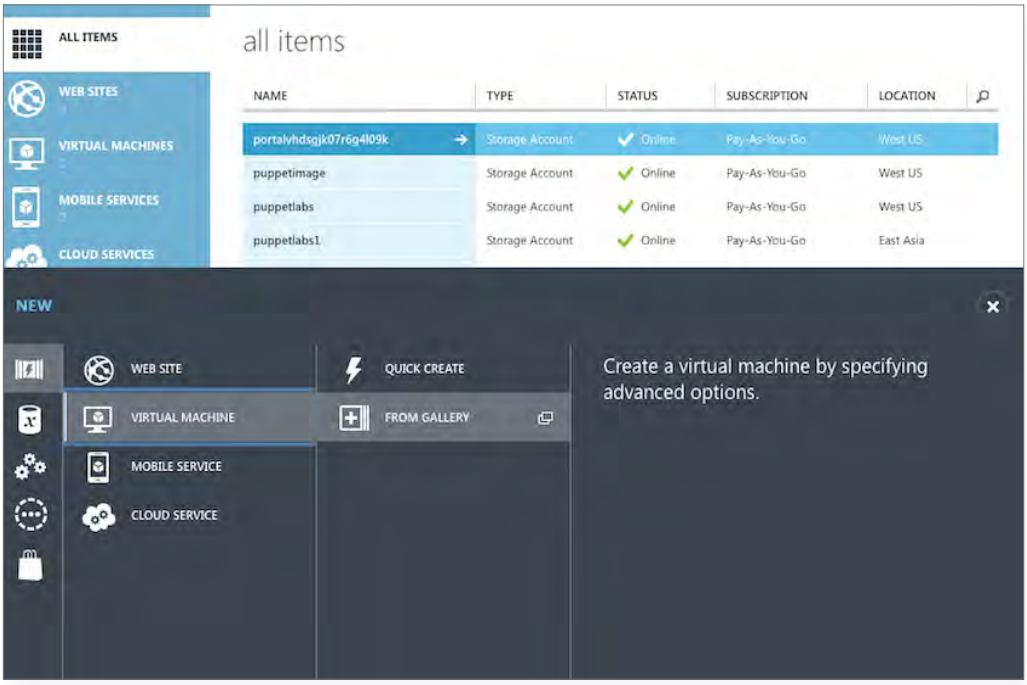
Puppet Enterprise is now offered in the Microsft Azure gallery, enabling you to quickly spin up VMs with the latest version of PE for evaluation.

This guide will help you use Azure to provision a puppet master and a puppet agent.

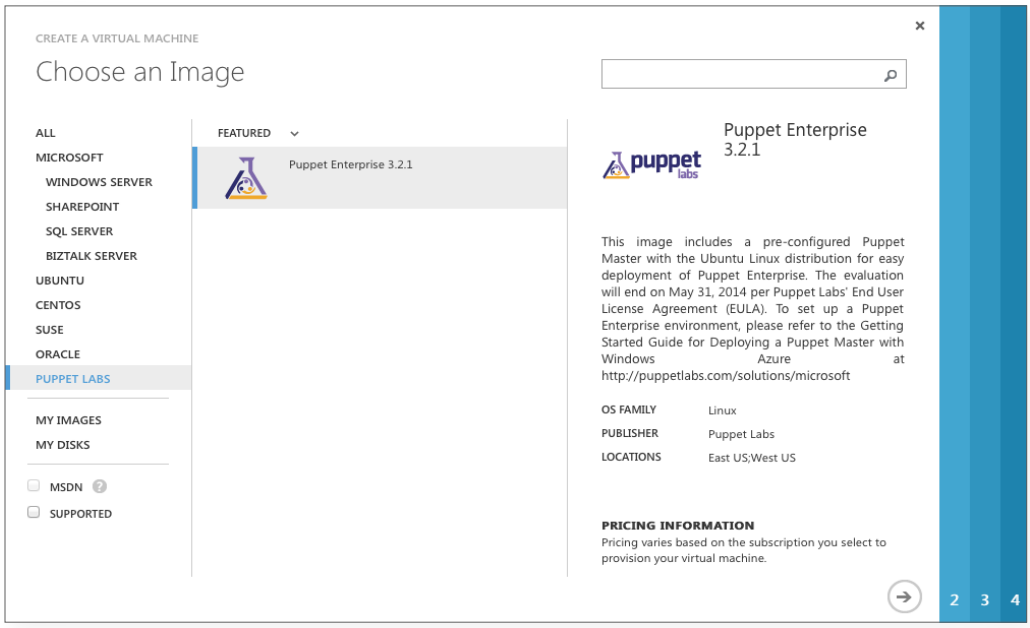
Install a Puppet Master

Before you get started, you should have created an ***Azure account***.

Step 1. Open the Azure gallery.
In the Azure portal, click **NEW** and then click **COMPUTE** -> **VIRTUAL MACHINE** -> **FROM GALLERY.**



Step 2. Choose the PE image.
Click **PUPPET LABS -> Puppet Enterprise 3.2.1.**



Step 3. Configure your virtual machine.
Provide the following information on the first config page. Click the arrow at the bottom of the page when you're done.

VIRTUAL MACHINE NAME: Can be any name of 3-15 characters, consisting of letters, numbers and hyphens. Letters must be lower-case; the SSL certificate won't be created if your name contains upper-case letters.

TIER: Choose **Standard**.

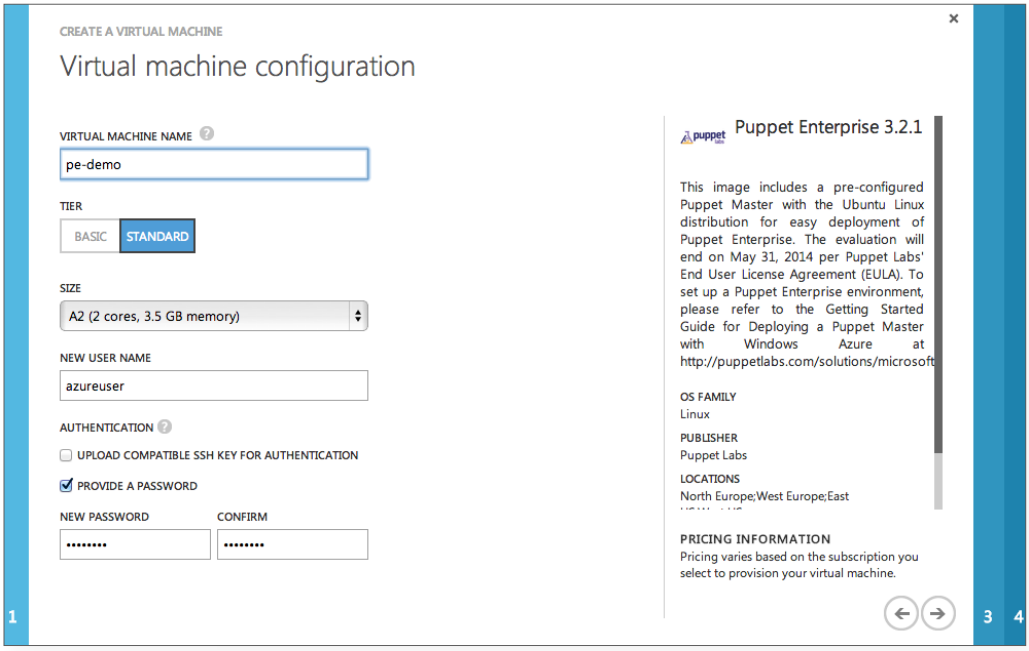
SIZE: Ensure that at least **A2 (2 cores, 3.5 GB memory)** is selected. You can bump up the size, but it's not necessary and will increase the cost. A smaller size will not provide enough power and the VM will underperform.

NEW USER NAME: Can be anything you want. This is basically your administrative account for accessing the machine.

AUTHENTICATION: You can choose to upload your SSH key, or to provide a password.

For these steps, we use a password: Click **PROVIDE A PASSWORD** and type one in.

If you want to use an SSH key, follow the steps to get a required X509 certificate described under ***SSH Key Generation***.



Step 4a. Continue to configure your VM.
On this page, you don't have to make many changes until you set ports in the **Endpoints** section. That's described in the next step.

CLOUD SERVICE: Always choose **Create a new cloud service**. This is key, because you don't want a puppet master behind a load balancer with any other machine. Otherwise, traffic will be redirected to other machines.

DNS NAME: This section should be prepopulated with your VM name and a suffix, such as cloudapp.net. This represents the public name of the VM.

REGION: Choose your region. By default, the region is prefilled with the region on your account.

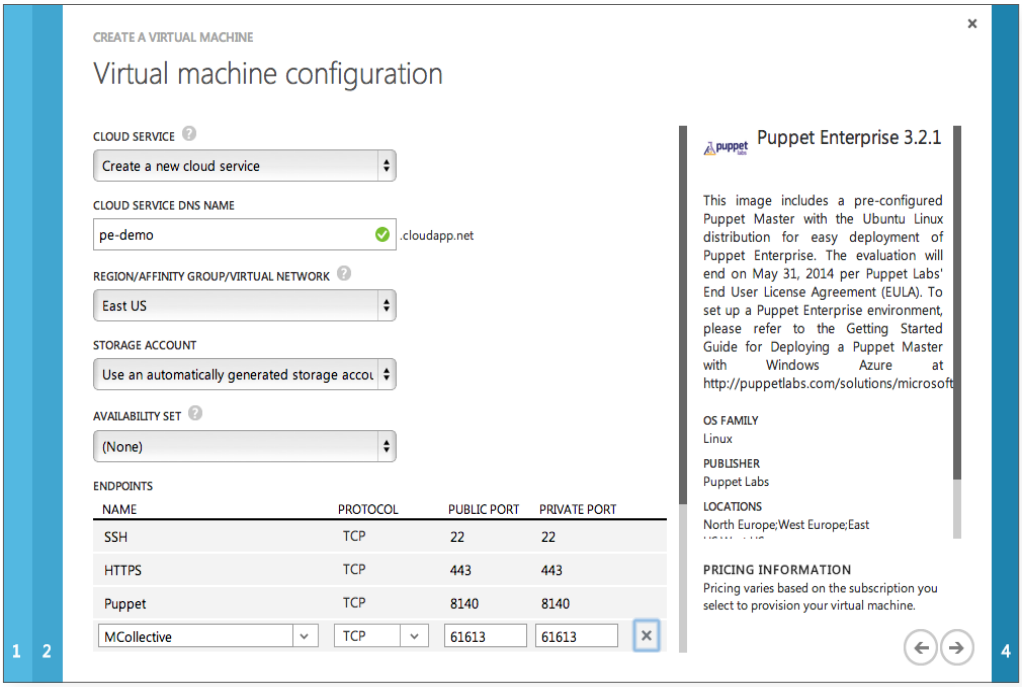
AVAILABILITY SET: Leave the default, **None**.

Tip: Take note of this fully qualified domain name (FQDN) for use when you create your puppet agent.

Step 4b. Configure your ports.
In the **ENDPOINTS** section, you configure your VM's ports. The SSH port is pre-set, because you can only run a puppet master on a Linux image.

Add the following mandatory ports. The first one can be selected from the drop-down list. Puppet and MCollective are the additional ports, and must be added manually.

HTTPS, to open port 443 for the PE console.
Puppet, to open port 8140 for puppet agents.
MCollective, to open port 61613 for MCollective.



Step 5. Create the virtual machine.
After you add the ports, click the arrow to move to the next page. Then, click the check mark at the bottom of the page.

The provisioning process begins. In your Azure window, you can see the name of the new VM in the list of VMs, with the status "Starting." The PE install takes several minutes. Eventually, you can hover on the name of the VM and an arrow will appear. Click the arrow to get details on the VM, including how many cores it has, what hardware is provisioned, and SSH information.

A note about DNS settings: Azure provides an internal and an external DNS name for your VM. For these steps, we use the external DNS name. This way, you can connect to puppet agents outside the Azure network, a common scenario. Doing so, however, entails opening two additional ports in the ENDPOINTS area.

Note: In the next section, you check to make sure your puppet master installed successfully. If not, check out the log that's located here: /var/log/upstart/puppet-init.log

Log Into the PE Console

Before you create the puppet agent VM, you need the password that's generated when the puppet master is provisioned. This is used to access the PE console, a web GUI, that enables you to manage node requests, assign classes to nodes, trigger puppet runs, and much more. For more information, see the ***PE console section of the online documentation.***

In this section, you get the password, and then you access the console.

Step 1. Open the PE console.
In a browser window, type "https://<public DNS name>".
For example: "https://pe-demo.cloudapp.net".

This is a good way to find out when your PE VM has been created. You'll know it's ready when you see the login for the console (see image). The process can take around 5-10 minutes.

Step 2. SSH to the VM you just created.
After the console login appears, indicating that the VM is ready, you can get the information you need to log in.

In your CLI, run the command `ssh <username>@<public DNS name>`.
For example: `ssh azureuser@pe-demo.cloudapp.net`.

Step 3. Answer the prompts.
The first time you SSH to the puppet master you will be asked to accept the remote host's public key. The message says that the authenticity of the host cannot be established, and asks if you're sure you want to connect anyway.

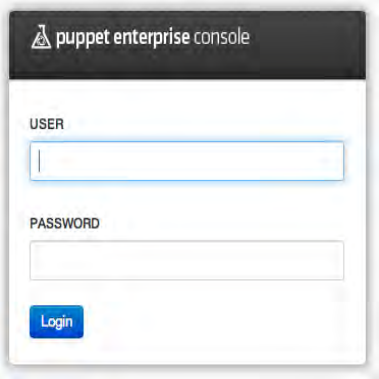
Type "**yes**". Then, type in the password you created when you set up the PE VM.

Step 4. Get the PE console user name.
Run `sudo grep 'auth_user_email' /etc/puppetlabs/installer/answers.install`.
Then, locate the user name in the answer file at this setting:
`q_puppet_enterpriseconsole_auth_user_email`.

The user name is `admin@<VM name>.cloudapp.net`.
For example, `admin@pe-demo.cloudapp.net`.

Copy the user name for use in step 6.

Note: If you forget to open a port when you're setting up the puppet master, you can do it afterwards: on the Azure dashboard, click the arrow on the puppet master VM, click ENDPOINTS, click Add, choose standalone endpoint, and select or create the port you want.



Step 5. Get the console password.
Run `sudo grep 'auth_password' /etc/puppetlabs/installer/database_info.install`
Locate the setting, `q_puppet_enterpriseconsole_auth_password`, which has the password appended to it.

It looks similar to this: `q_puppet_enterpriseconsole_auth_password=thos9Greu`.

Copy the password for use in step 6.

Note: The password will not be available until the master install is complete. You might have to refresh the brower window a few times until you see it.

Step 6. Open the PE console.
In the PE console login, type in your user name and password to log into the console. The first time you connect to the console, you get an SSL security warning. It's safe to proceed. For more information, see the ***PE documentation.***

Because you haven't set up any agents yet, there's not much information here.
Click **Node** and you'll see the puppet master node listed, along with its private DNS name.

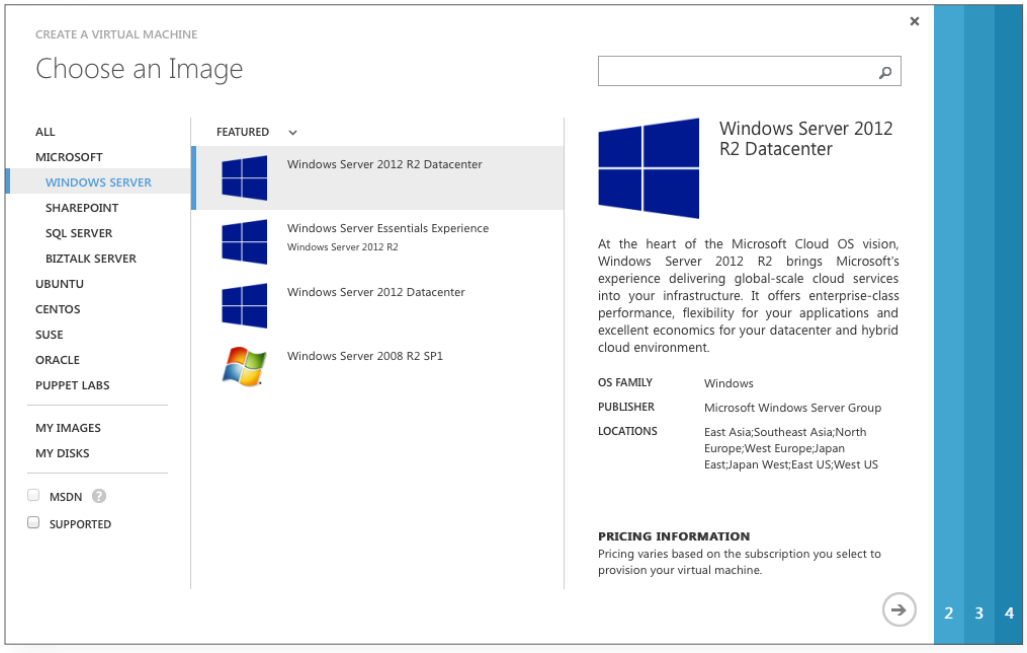
After you add an agent in the next section, you'll return to the console.

Install a Puppet Agent

These steps show how to bootstrap a puppet agent with an Azure VM at provision time. The steps are similar to configuring the PE VM.

Step 1. Launch the VM workflow.
In the Azure portal, click **NEW** and then click **COMPUTE** -> **VIRTUAL MACHINE** -> **FROM GALLERY**.

Step 2. Choose an image, in this case a Windows node, from the list.
Click **MICROSOFT** -> **WINDOW SERVER** -> **Windows Server 2013 R2 Datacenter.**



Step 3. Begin configuring the VM.
Provide the following information and then click the arrow to go to the next page.

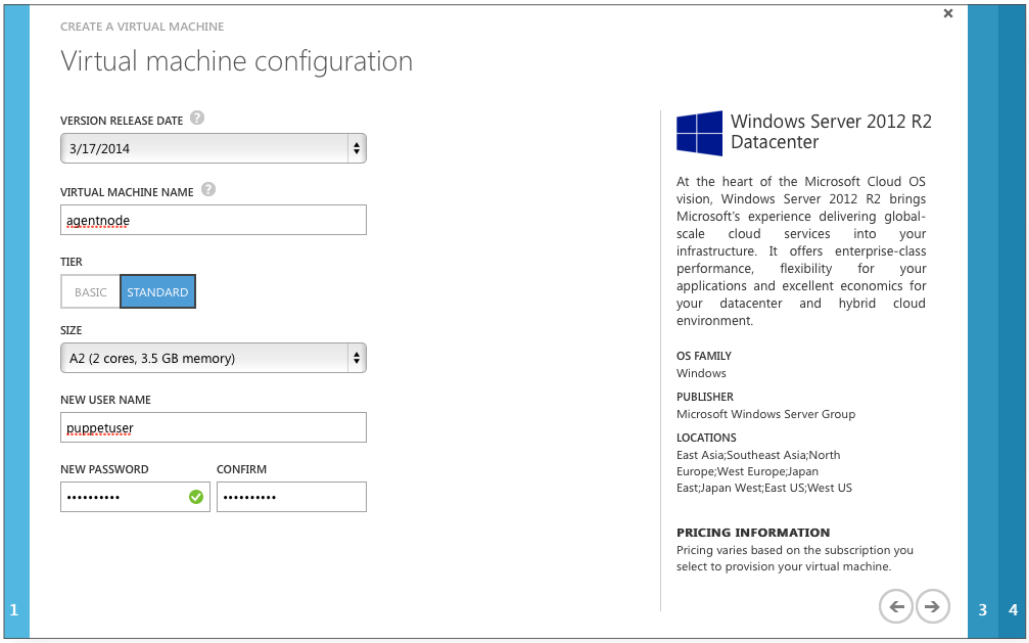
VIRTUAL MACHINE NAME: Name the machine. Remember to use all lower-case letters. Can be something like "agentnode".

TIER: choose **Standard**.

SIZE: Choose whatever size will suit this VM's anticipated workload.

NEW USER NAME: Can be anything you want.

NEW PASSWORD: This is also your choice.



Step 4. Continue configuration.
Provide the following information, and then click the next button again.

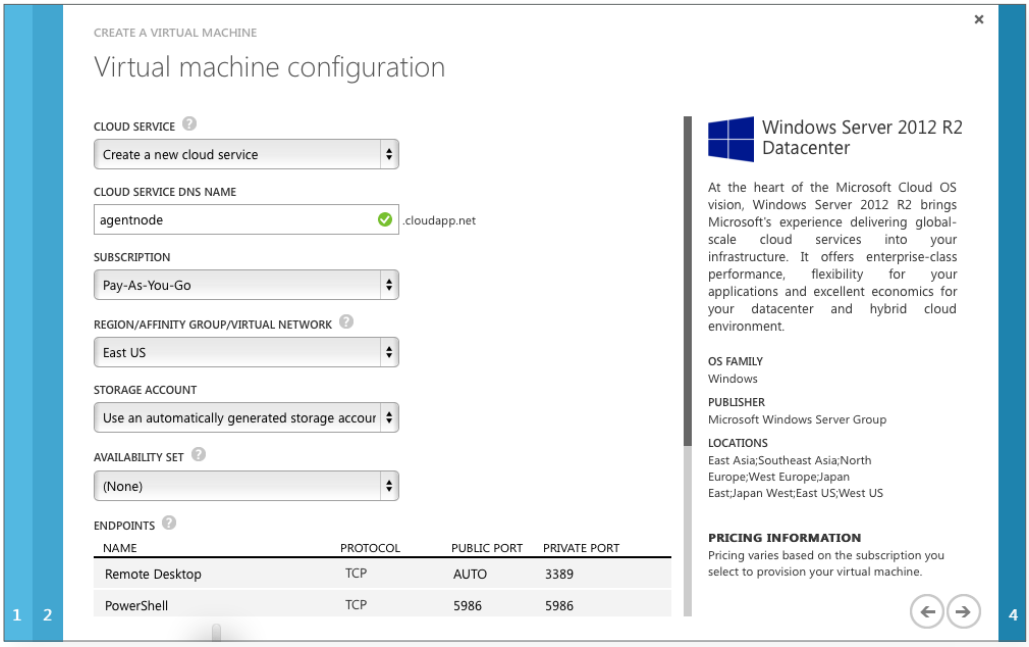
CLOUD SERVICE: Can leave at default.

DNS NAME: This section should be prepopulated with your VM name and a suffix, such as cloudapp.net.

REGION: Choose your region.
The region is prefilled with the default region configured for your Azure account.

AVAILABILITY SET: Leave the default, **None**.

ENDPOINTS: Use the defaults, Remote Desktop and PowerShell.



Step 5. Choose the VM agent to enable the Puppet Enterprise extension.
Select the **Install a VM agent** check box. Then, under **OPTIONAL EXTENSIONS** select the **Puppet Enterprise Agent** checkbox.

Step 6. Provide the FQDN of the puppet master you set up previously.
In the **PUPPET MASTER SERVER** box, type in the DNS for the puppet master VM you created previously.

Then, click the check mark at the bottom of the page to start provisioning the agent VM.

Once the VM is provisioned, the puppet agent is installed and it registers with the puppet master you designated. This process can also take several minutes.

When it's ready, the agent sends a certificate request to the master.

Approve the Agent Node Request

In this section, you view the puppet agent in the PE console, and accept the agent's Certificate Signing Request (CSR).

Step 1. Open the PE console as you did previously.

Step 2. Locate and approve the agent request.
At the top of the console window, click **node requests** to open the node requests view. There should only be one request in the list. Click **Accept**.
If your agent isn't listed in the **node requests** page, see the following section.

Establish a Remote Connection Using RDP

The following steps describe what to do if your puppet agent isn't connecting to your puppet master.

Step 1. Look up the agent's public FQDN.
In Azure, click the agent VM, and then click **DASHBOARD.** Note the DNS Name in the **quick glance** area of the dashboard.

Step 2. Find the dynamically assigned RDP port number.
Click **ENDPOINTS** and note the **PUBLIC PORT** for Remote Desktop.

Step 3. From Windows, run Remote Desktop.
For **Computer**, type <DNS>:<public port>.
For example, "win-agent.cloudapp.net:58859".

Step 4. RDP to the agent.
Under **Puppet Enterprise**, right-click **Start Command Prompt** and click **Run as administrator**. In PowerShell, type `puppet agent --configprint server`.

Is your puppet agent trying to connect to the correct puppet master node?

Now that you've completed this Getting Started Guide, check out the ***Puppet Enterprise documentation*** to learn more about the automation and configuration capabilities of PE.

926 NW 13th Ave #210
Portland, OR 97209
877.575.9775
www.puppetlabs.com

About Puppet Labs

Puppet Labs, Inc. is the leader in IT automation. Puppet Labs' software provides system administrators the operational agility, efficiency and insight they need to proactively manage dynamic infrastructure, scaling from tens of servers to thousands, on-premise or in the cloud. Thousands of the world's leading organizations are using Puppet Labs' software to configure and manage their IT infrastructure, including Citrix, eBay, NYSE, Match.com, Oracle/Sun, Shopzilla, and Zynga. Now numbering more than 140 employees and based in Portland, Oregon, Puppet Labs is backed by investors Kleiner Perkins Caufield & Byers, Google Ventures, VMware, Cisco, True Ventures, Radar Partners, and Emerson Street Partners.