Cloud Management - Self-Service

[PDF generated September 09 2025. For all recent updates please see the Nutanix Bible releases notes located at https://nutanixbible.com/release_notes.html. Disclaimer: Downloaded PDFs may not always contain the latest information.]

In the following sections we will cover the key features of Nutanix Self-Service. In particular:

- Blueprint-centric infrastructure and application design
- Automation with Self-Service endpoints and runbooks
- Pre-built application deployment with the Self-Service marketplace
- NCM Self-Service integration with Prism Central projects
- NCM Self-Service Policies for infrastructure governance
- NCM Self-Service DSL (Domain Specific Language)

Capabilities Summary

NCM Self-Service is a multi-cloud management framework that offers the following key benefits.

- Enhanced efficiency and agility along with potential reduction in human error
- Multi-cloud orchestration across Nutanix AHV, VMware (Nutanix and non-Nutanix), AWS, GCP and Microsoft Azure
- Application management via the NCM Self-Service Marketplace's pre-seeded blueprints as well as the ability to publish client-created blueprints to the marketplace
- Cost Governance integration to analyze and visualize overall resource utilization
- Kubernetes integration for pod- and container-based deployments
- Self-Service DSL (Domain Specific Language) for Infrastructure-as-Code blueprint design

Blueprint-centric Application Design

NCM Self-Service's blueprint- and infrastructure-centric application design allows automation and deployment administrators to quickly prototype and deploy infrastructure and applications of any design. Within NCM Self-Service blueprints, two blueprint types are available:

- **Single VM Blueprint**: An application or service consisting of a single VM, along with its associated deployment scripts and dependencies. Dependencies includes items such as hardware configuration, operating system types and versions and tasks that a deployment will need to step through before completion. Official Single VM Blueprint documentation on Nutanix Support Portal.
- **Multi VM/Pod Blueprint**: An application or service consisting of one or more VMs, along with each VM's associated deployment scripts and dependencies. Similar to Single VM blueprints in overall concept, Multi VM/Pod Blueprints are different in that relationships between an application's VMs can be defined during deployment ensuring, for example, that a database is deployed before the application that depends on that database. In addition to the core application components, Multi VM/Pod Blueprints include configuration for scaling and lifecycle management that allow administrators or triggers to manually or automatically adjust configuration based on application demand. In addition to VM deployments, Multi VM/Pod Blueprints support the deployment of pod-based applications in a Kubernetes environment e.g. Deployments, Containers and Services. Multi VM/Pod Blueprint documentation on Nutanix Support Portal.

Deployment Locations

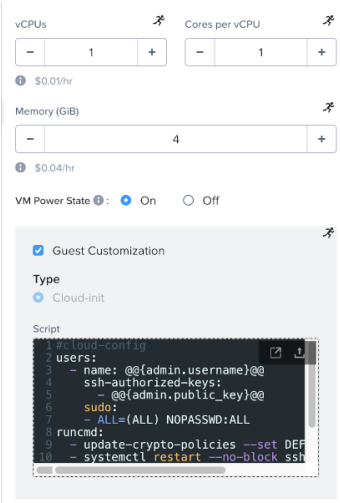
NCM Self-Service blueprints support deployment to various on-premises and cloud locations:

- On-premises with Nutanix AHV or VMware vSphere
- Cloud locations including Nutanix Cloud Clusters (NC2), Amazon AWS, Microsoft Azure, Google Cloud Platform and Kubernetes

Blueprint Example

The diagram shows an example of a Multi VM/Pod Blueprint. Specifically, the following configuration items are visible:

- A single virtual machine's hardware configuration i.e. vCPUs, cores per vCPU, VM RAM in GiB and, in this example, a Linux Cloud-Init spec that will be applied to the deployed VM. This particular Cloud-Init specification also includes the SSH keys that will permit SSH login to the deployed VM.



Tasks

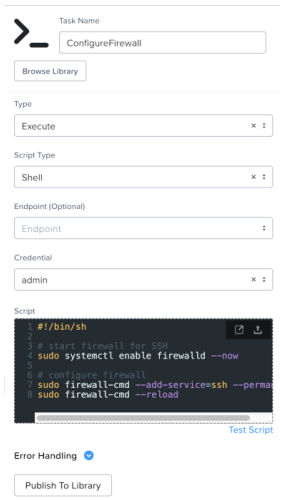
NCM Self-Service blueprints carry out deployment activities via tasks. A task contains individual steps and actions required to complete a deployment or, in the case of day-2 operations, steps and actions to customize an existing deployment. Tasks can be executed either sequentially or in parallel, depending on the application's requirements.

For example, a typical VM deployment includes a "Package Install" step, containing tasks that run in order to customize the VM's configuration. These tasks run after the VM's base operating system has been deployed and after VM customization with Cloud-Init (Linux) or Sysprep (Windows) has completed.

The example shown here is the same VM in the above screenshot; during the Package Install step, two scripts will be run:

- A task named "BaseVMConfiguration": A shell task during which the Linux OS package cache and system packages will be updated. This step includes the installation of the Linux package named "firewalld".
- A task named "ConfigureFirewall": A shell task during which the firewalld service will be configured, then started.

Looking deeper into this example, observe the task details that allow selection of where the script will run (endpoint), the task type (Execute), the script type (Shell) and the credential that will be used to run the script (admin).



Blueprint task scripts can be easily saved to the built-in Self-Service Library, allowing quick and easy re-use across multiple blueprints.

Brownfield Applications

In the context of NCM Self-Service, a brownfield application is a collection of services (e.g. VMs) that are not yet managed by Self-Service. Before Self-Service can communicate with those services, the application must be imported into Self-Service as a brownfield application.

Brownfield application key points:

- Privileges: Administrative privilege are required to import a brownfield application
- Brownfield applications do not support snapshot and restore capabilities

See Brownfield Apps in Self-Service for detailed information.

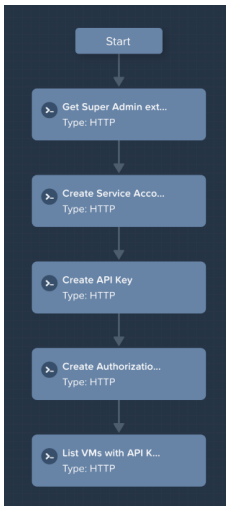
Runbook Automation

NCM Self-Service supports manual- and trigger-based activity via Runbooks. In the concept of NCM Self-Service, a Runbook is defined as a set of steps that can be executed sequentially in order to achieve a specific goal. For example, a manually-executed runbook could be created to complete the following steps:

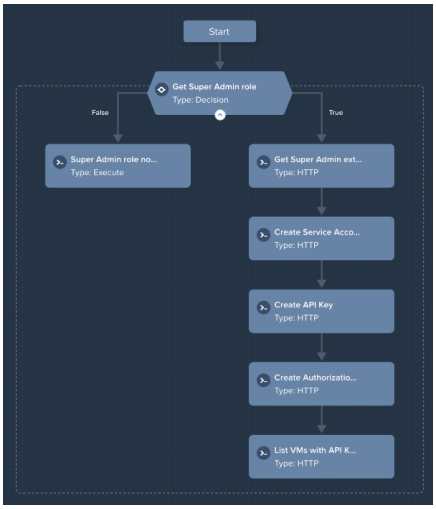
- Use the Nutanix v4 APIs to retrieve the identifier of a Nutanix IAM role named "Super Admin"
- Create a new user of account service account and associate that new user with the "Super Admin" role retrieved in the previous step
- Create a unique API Key for the new user
- Create a Nutanix IAM Authorization Policy for the new user that allows permission to create new VMs and list existing VMs

This real-world runbook is an example of steps that could be run in response to Nutanix X-Play detecting the creation of a new virtual machine. By combining these two features, the manual input required to create an authentication and authorization scheme can be significantly reduced.

The example shown below is this exact runbook, with the addition of a final testing step that verifies all previous steps were successful.



As an additional example, the runbook shown here completes the same steps as above but with an additional first task. Specifically, this runbook will retrieve the identifier of our "Super Admin" role but will then carry out a check to make sure that role was found. If the role was not found, the runbook gracefully exits without making any changes. If the role was found, the runbook steps through the remaining tasks as designed.



Endpoints

Each step within an NCM Self-Service runbook can be associated with a specific endpoint. In the context of running specific steps within a runbook execution, this enables the runbook designer to run specific steps on different systems.

For example, a step that sends a REST API request could send the initial request to Prism Central instance "a", save the results of that request into output variables, then run subsequent steps on Prism Central instance "b". This allows a runbook to operate in environments with a single Prism Central instance, in environments with multiple Prism Central instances or in environments with arbitrary/user-configured endpoints. There is no restriction on the type of endpoint that can be configured, provided NCM Self-Service can login to that endpoint.

Additional examples can include running a PowerShell or Python script on a jump host in environments where security posture dictate script execution must take place on specific hosts only.

As with NCM Self-Service blueprints, scripts can be quickly saved to the NCM Self-Service Library for easy re-use, and loaded directly from the library when a pre-written script is required.

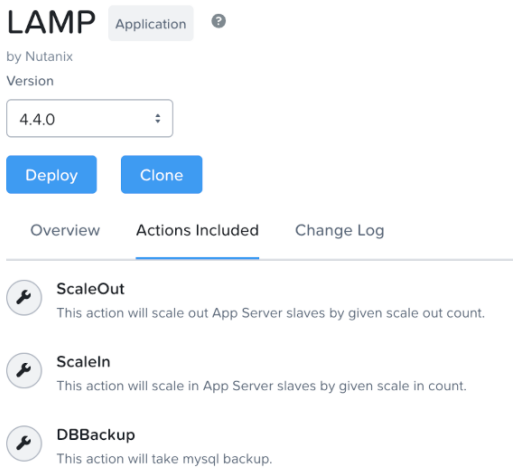
Marketplace Application Deployment

Single and Multi VM blueprints allow the creation of highly-customized applications based on specific criteria. However, when an application design is complete it can then be made available to deployment administrators via the NCM Self-Service Marketplace.

The NCM Self-Service Marketplace is a way of deploying pre-configured applications complete with deployment approval requests, analysis of a deployment's procedures before any changes are made and, when necessary, the ability to deny a deployment request when appropriate.

Out-of-the-box, a number of pre-packaged marketplace applications are available based on best-practice deployment methodologies. These include familiar and popular applications such as LAMP Stack (Linux, Apache, MySQL, PHP), Jenkins, GitLab and many others. All applications i.e. both built-in and custom are version-controlled to ensure a deployed application meets an environment's criteria before any changes are made.

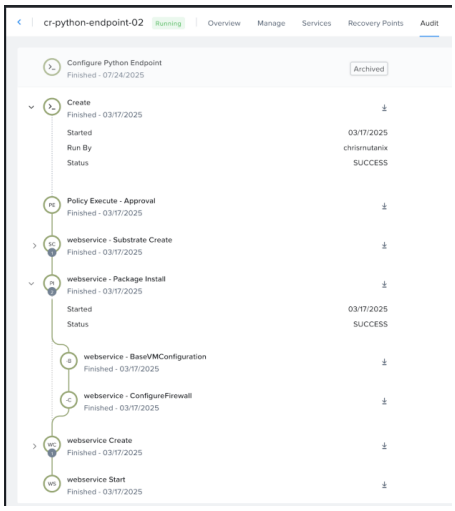
The built-in LAMP Stack application is shown here, including a description of some of the included actions. In the context of an NCM application or blueprint, an action is either a step that runs during deployment or a post-deployment step that can be executed on-demand or based on a specific trigger.



Depending on the permissions given to the user requesting the deployment, the application install will either proceed as normal or be queued for approval by an authorized user.

All NCM Self-Service blueprints, including Single VM Blueprints, can be published to the NCM Self-Service marketplace. This allows extensive customisation of IT-managed applications, during both test/development and production deployment phases. As part of the publication process, the approval chain is included; an application must be approved for publication to the marketplace before any user can request its deployment.

The following screenshot shows the ability to monitor a blueprint deployment directly from the NCM Self-Service UI. This deployment view is available in both manual application launch from the blueprint designer and the audit tab for marketplace-launched applications.

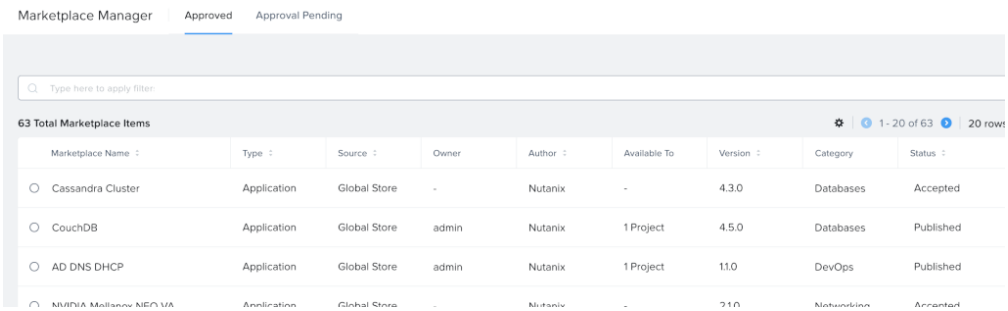


Built-in Application Blueprints

Some of the built-in, pre-built applications available are:

- LAMP Stack (Linux, Apache, MySQL, PHP)
- CouchDB (open source database)
- GitLab (web-based Git repository manager)
- OpenLDAP (open source LDAP implementation)
- Redis Cluster (in-memory structure store)
- Jenkins (open source automation server)

All Self-Service Marketplace administrative operations, including approvals, are carried out via the Marketplace Manager (shown below).

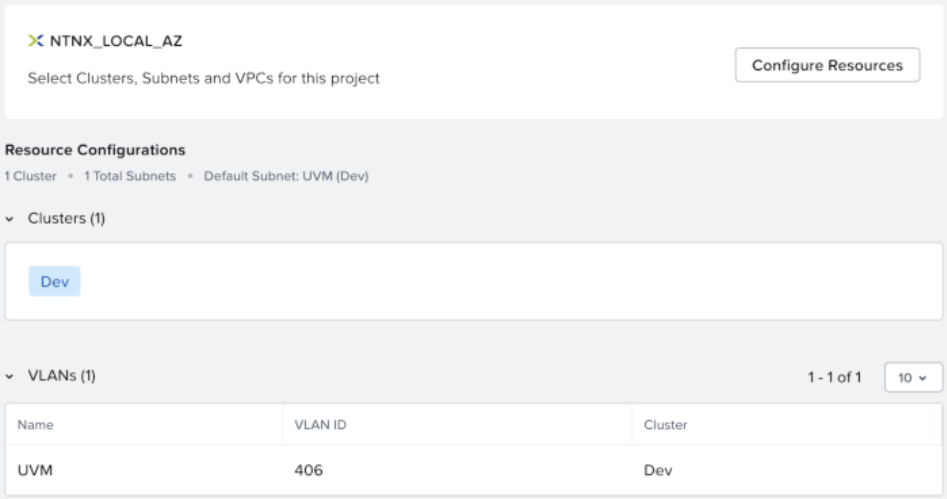


Self-Service integration with Prism Central Projects

NCM Self-Service makes extensive use of Prism Central Projects. A project is a collection of components and configurations that define how and where a VM or application can be deployed. For example:

- Users and groups
- Infrastructure
  - Clusters that will host VMs and application components
  - Associated networking resources e.g. subnets
  - Policies and quotes via the NCM Self-Service Policy Engine
  - Linked services such as Nutanix Database Services (NDB)

The following screenshot demonstrates the selection of infrastructure during project configuration. By making these selections during project creation or edit, an administrator can restrict deployments to specific clusters and connections to the specific subnets in that cluster.

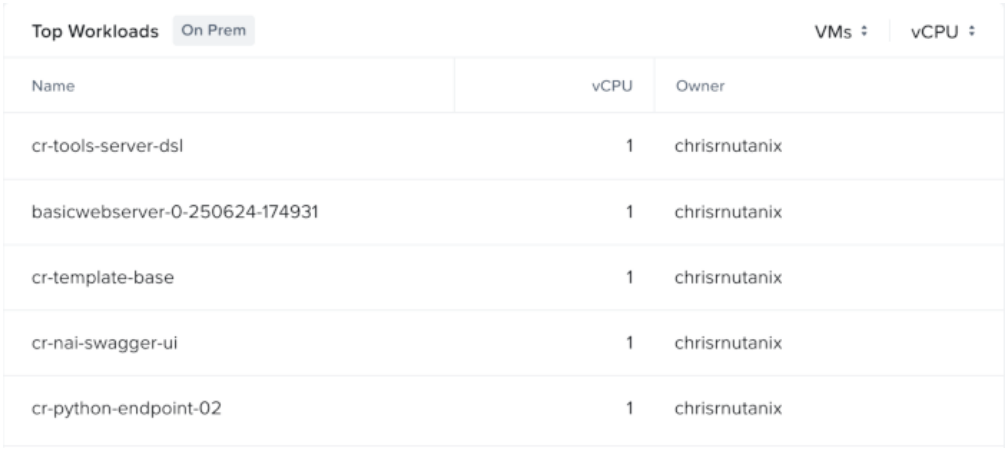


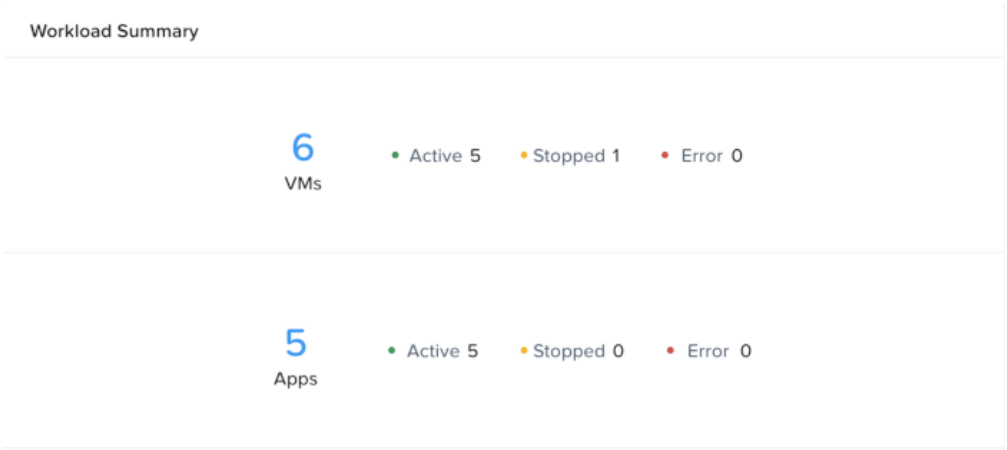
The integration of Prism Central projects into NCM Self-Service provides the ability to control which users are permitted to carry out specific tasks, *where* those tasks can be carried out and limits of the number of resources a user's deployment may consume. This approach to quota management provides a robust and highly configurable approach to tenant management within an NCM environment.

Project Analysis

Because of this deep integration between NCM Self-Service and Prism Central projects, analysis of workload allocations within a project is easily visible. A user can quickly see resource utilization of a deployed application within a project, the number of deployed VMs for a specific project as well as quota consumption for that project.

The following screenshots demonstrate a) the top workloads for the selected project and b) the workload summary in terms of VM deployments vs deployed applications, respectively.





Self-Service Policies

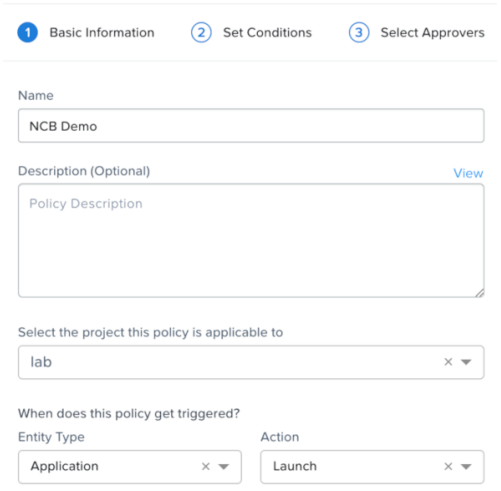
NCM Self-Service provides various policy-based capabilities. A scheduler policy allows Self-Service administrators to create scheduled tasks. For example, a scheduled task to stop an application at a certain time after hours and restart the application the following business day. A scheduled task can also be used to run an NCM Self-Service Runbook, allowing administrators to complete a defined set of steps on a chosen endpoint at a specific time of day.

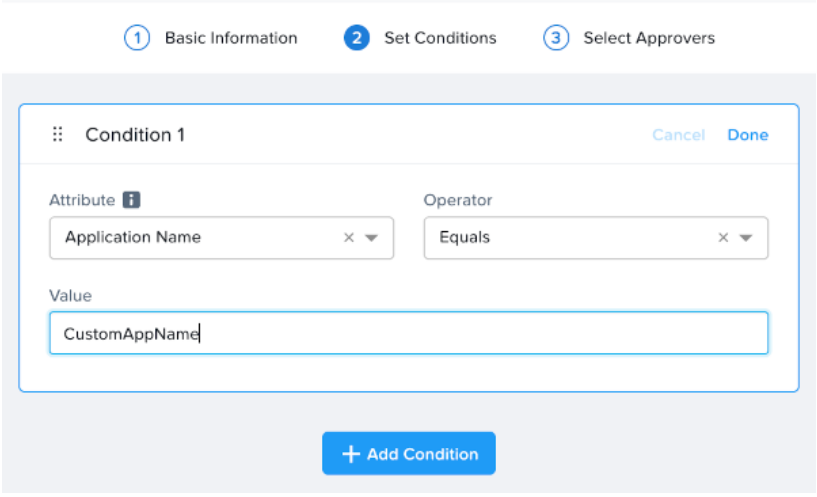
Self-Service policy capabilities also provide detailed control over application approvals. This level of governance ties directly into the Self-Service Marketplace feature outlined above and defines which applications or actions require approval before being run.

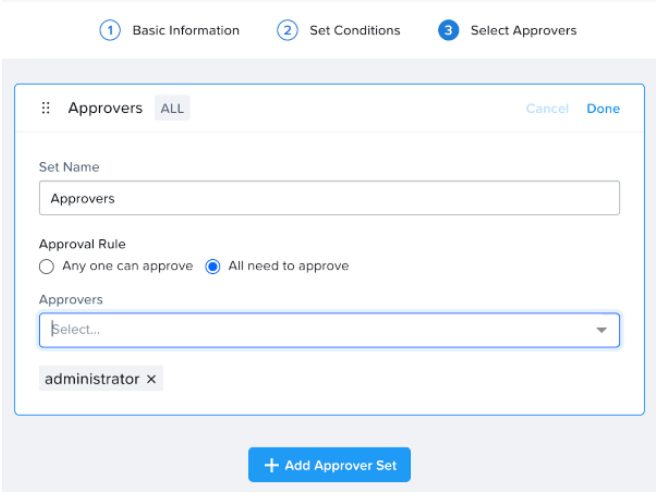
Combinations of scheduler and approvals policies can help prevent cost-associated issues, especially relating to infrastructure resources running in environments with time-based charges.

All NCM Self-Service policies are managed through the Policy Engine, handled directly within the NCM Self-Service UI.

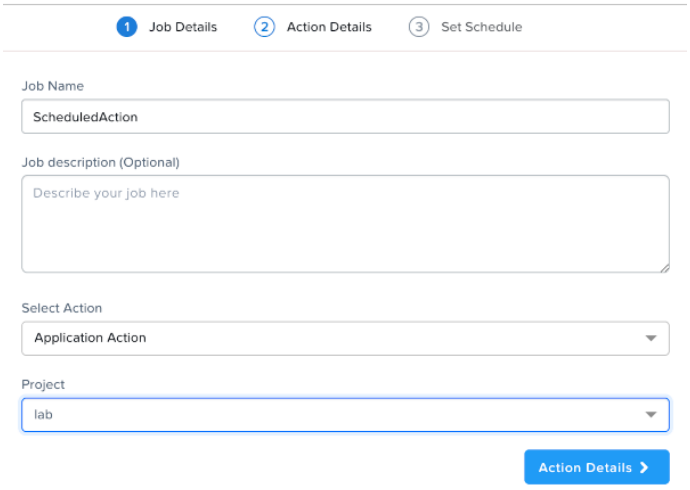
The following screenshots show an example of approval policy creation. In this example, approval must be completed by all selected users ("administrator") whenever an application named "CustomApp" is launched.

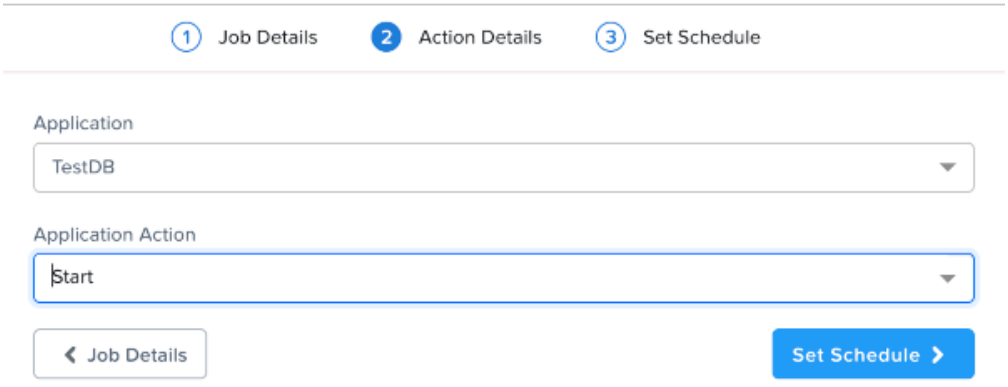


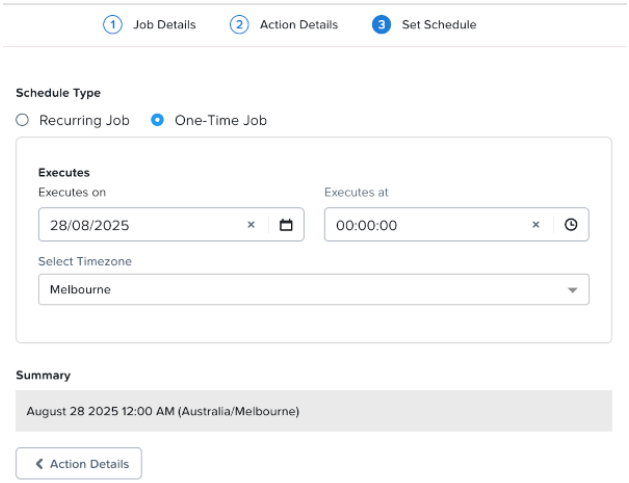




Similarly, the following screenshots demonstrate the execution of an application action at a specific time of day. Whilst this example shows a single-run, the same interface is used for recurring schedules.







NCM Self-Service DSL (Domain Specific Language)

In addition to the extensive management capabilities available in the Self-Service UI, NCM Self-Service provides an extremely broad, developer friendly DSL. A DSL, or Domain Specific Language allows developers and scripting administrators to leverage Self-Service using standard programmatic approaches. The NCM Self-Service DSL specifically is an open-source Python language enabling the programmatic control of Self-Service features directly from your Python scripts.

Some Self-Service DSL highlights include:

- **Ease of use**: Straight-forward syntax enabling the easy consumption of key NCM Self-Service features such as blueprint creation, application launch and day-2 operations
- **Platform agnostic**: Python is supported on a wide range of operating systems, making the Self-Service DSL's code usable in an equally wide range of environments
- **Portability**: A complete, DSL-based deployment can be packaged and moved between environments with minimal effort, or stored in source control systems such as GitHub or GitLab