Chapter 1 : An Introduction to Fedora Linux

Introducing Fedora: Desktop Linux is here to tell you that this is simply not the case. Fedora, and Linux in general, have become very easy for everyday home and business users to install and use. Now you can find out for yourself, in just one weekend, how easy it is to get started using Fedora and discover that there is life beyond Windows.

When major corporations are moving towards open architecture by using open source tools and even pushing their internal projects into open source, it makes your contributions especially worthy. But before starting with contributing, many people face the same common set of questions. How they can start, how should they introduce themselves in the community, and where they can contribute. During the planning phase , I got in touch with Sumantro , who is himself an open source enthusiast and contributing to various open source projects including the Fedora Project. With his help, we planned the agenda for the session and gathered the resources to conduct the session. Answer these questions Bring up new people in the open source arena Show where they can contribute, learn and make an impact Starting the day in Delhi The session started with small questions. People who have tried their hands on contributing to open source shared what problems they faced during their journey. The answers to the questions ranged from having issues working with the codebase of the projects to problems in figuring out where to start from. During the session, the main focus was to make the participants aware about what FOSS is and how it is beneficial to the community. The common misconception about freeware and FOSS being the same was also cleared during the session. During the session, a brief overview was presented to the participants about how they can start with their open source journey. We walked them through from identifying the project where they want to contribute to sending their introduction mails in the mailing lists of the project. The session moved on with the topic about where the participants can contribute and what areas of contribution they can work in both technical and non-technical. Introducing Fedora After the introductory session on FOSS, we went ahead with our agenda and introduced the Fedora Project and the community behind it: The participants were guided upon how they can create their identity on the Fedora Project by signing up on FAS. They could then use that identity to get access to various Fedora applications and resources. The session on Fedora moved on with the introduction on how the contributors can get to the mailing list and introduce themselves to the community. There, they can get help about starting their contributions. The main focus during the session on Fedora was to introduce the participants to the Fedora Quality Assurance QA team and release validation testing. Extending on the basic idea, we introduced Bodhi and package testing. Through a live demonstration, participants learned how they can start with package testing. The demo consisted of how to log into Bodhi using FAS and then enabling the updates-testing repository on Fedora to get the packages in testing. An overview of the karma system was provided to the participants where they were told about what karma points mean and how they should give karma. The session proceeded with an overview of release validation and why it is important. The different development channels in Fedora, like Rawhide and branched, were introduced to the participants and what they mean. A demo of release validation using relval was provided to the participants. Introducing Git and more questions The next session for the event was focused on getting started with Git. Git is a version control system used by many individuals and corporations to manage their source code. It also keeps track of the changes made by other developers. During this session, participants were introduced with a basic Git workflow. How they initialize a Git repository, add a remote repository, and pull the project source code. Participants guided to making their first Git commits and how it all works. This included covering the associated benefits of this type of system. Moving on with the Git session, participants were introduced to how various open source projects use version control systems like Git to manage their source code and accept contributions. The event ended with an open question-and-answer session. Participants asked a variety of questions regarding open source projects. These questions consisted of things like availability of paid opportunities in open source, competitions in open source, and more. Answering these questions, participants learned about programs like Google Summer of Code , various conferences that are organized by the open source projects, and the recognition model used by these projects.

Chapter 2 : Team Silverblue â€” Elsewhere

Swift is a general-purpose programming language built using a modern approach to safety, performance, and software design patterns. It aims to be the best language for a variety of programming projects, ranging from systems programming to desktop applications and scaling up to cloud services. Read.

Do you want to learn more about the history behind Hubs? Hubs features Here are some key features in Hubs right now. Development is continuing, so these features will change over time. Group hubs are associated with respective FAS groups. It lists a feed for the team, as well as their configured widgets. The widgets are listed in the order they were added with their index values. You can see the values in the populate. Also, it becomes easier to get insights from the other teams on a particular task or project. User hubs are configurable. The various sized rectangular cards on a hub page are called widgets. Every user hub page features a Contact-Info widget, for example. Work-in-progress features The team is currently working on more features, including: The User hub pages will feature a vertical bookmark bar. Hubs will offer suggestions based on the frequency of visits to each hub page. Using this widget, a user can chat with developers and not worry about losing logs. Want to hack on Hubs? This post also provides a glossary of Hubs-related terms. Getting Help Of course we have a number of ways you can reach the Hubs team. We have weekly meetings every Tuesday at Never used IRC before or a little bit intimidated? You have to subscribe to the list if you want to post. You can subscribe to the Fedora Hubs mailing list at this web page. To subscribe to the list, use any email address you prefer. Hope to see you soon, hacking on Hubs!

Chapter 3 : Introduction :: Fedora Docs Site

The goal of the Fedora Hubs project is to provide a consistent contributor experience across all Fedora teams. Hubs serve as an "intranet" for the Fedora Project. The many different projects in Fedora each have different processes and workflows. Hubs will be a single place where.

In Fedora 27, we have reached a point where we feel comfortable inviting other developers and enthusiasts to try it out and even make it their daily driver. Read on to discover what Fedora Atomic Workstation is, what its benefits are, and how you can get started today! What is Atomic Workstation? However, Atomic Workstation is geared towards all the same use cases that the regular Workstation Edition is meant to fulfill. Though there are some differences between the two beyond the update model. In an Atomic Workstation, desktop applications are shipped and run as flatpaks , and development mostly happens inside containers. For example, you may have a pet container with your development environment set up as well as an oc cluster up OpenShift cluster to develop server applications. Why should I use Atomic Workstation? So what are the advantages of this strategy? Many of the reasons below are shared with Atomic Host. Though I will try to give a more Workstation-centric point-of-view. Transactional Updates The main reason to use Atomic Workstation of course is transactional updates. This is as relevant in the server case as it is for desktops. Most people would consider their workstations to be the stereotypical pet system: Using an update model which greatly reduces risks of failures is thus well justified. I will not stay on this subject any longer since I assume readers of this blog are familiar with the benefits there and what pitfalls the Atomic model helps us avoid. Immutability and Isolation As discussed in a previous blog entry , all the great features of an OSTree-based system require immutability of the base OS. This is equally true in Atomic Workstation as it is in Atomic Host. This is a good thing, because 1 it protects you from accidental damage , and 2 it encourages a healthier workflow. On rpm-ostree-managed systems, all scriptlets are run on the server when composing the update. As mentioned lower down, we do also support layering additional RPMs; their scriptlets are run in a locked down container and cannot affect the running system. By running your applications in flatpaks and doing your development in containers, you not only help protect your base OS from harm but also minimize the number of packages standing between you and a successful boot. Basing your development workflow around pet containers could warrant a blog post of its own. There have been some great talks around that area at DevConf. We can, for example, do rpm-ostree install vim-enhanced libvirt-client to layer additional packages. All these changes can equally be tracked in rpm-ostree status and easily reverted. Upstream Testing Having updates delivered as concrete units allows the content provider to more easily test them, which translates into a more stable operating system for end users. This mostly means that testing is limited to the package-level. Note though that Atomic Workstation updates are not currently being gated on tests; we are still at the onset of these discussions. Similar functionality on dnf-managed systems does exist. The difference is that the OSTree model allows us to eliminate or minimize many of the issues involved in such a scheme. How can I get started with Atomic Workstation? You can either download the ISO and re-install from scratch, or you can convert your existing dnf-based system using the steps from this document.

Chapter 4 : FOSS Wave: Delhi, India â€“ Fedora Community Blog

The user desktop in Fedora 16 is very close to that of Fedora the changes are mostly transparent to the end user, affecting mostly in the inner workings. See our video review of Fedora The user desktop in Fedora 17 is indistinguishable from that of Fedora again the changes are mostly behind the scenes.

Chapter 5 : Introduce yourself to the Docs Project - Fedora Project Wiki

From The Community. Amazon Try Prime. Kindle Store.

DOWNLOAD PDF INTRODUCING FEDORA

Chapter 6 : LXer: Introducing Swift on Fedora

This guide covers installation of Fedora, a Linux distribution built on free and open source software. This manual helps you install Fedora on desktops, laptops, and servers.

Chapter 7 : Introducing my adVintage. | The Fedora Lounge

Introducing Fedora Today we are pleased to offer Fedora Server 24 for deployment on your Linode. The Fedora Project aims to provide the latest stable packages and approximately 13 months of.

Chapter 8 : Linode Blog Â» Introducing Fedora 24

The late Coach Earle Bruce was a long-time champion in the fight against Alzheimer's Disease. Over the course of 12 years, Coach hosted several events to raise money for a variety of funds and causes, with most efforts going to The Earle & Jean Bruce Alzheimer's Research Fund at The Ohio State University.

Chapter 9 : Get Fedora Server: the latest tech for your apps and services

Introduction to Linux - A Hands on Guide This guide was created as an overview of the Linux Operating System, geared toward new users as an exploration tour and getting started guide, with exercises at the end of each chapter.